SCHAUM'S ouTlines

全美经典 学习指导系列

Mathematica使用指南

〔美〕D. 尤金 著

邓建松 彭冉冉 译

- 750多道例题和给出解答的问题
- 介绍了广泛用于技术与科学计算的语言
- 优秀Mathematica和计算机数学教材的补充
- 理想的考前复习资料


科学出版社

麦格劳-希尔教育出版集团

(O-1516.0101)
责任编辑：陈玉琢

SCHAUM'S ouTlines

全球销量
超越3000万的
"全美经典学习指导系列"
是您的最佳
学习伴侣！

40年来最畅销的教辅系列
全美著名高校资深教授倾力之作
国内重点高校任课教师全力推荐并担当翻译
省时高效的学习辅导，全面详细的习题解答
迄今为止国内最全面的教辅系列
覆盖大学理工科专业

全美经典学习指导系列

概率和统计	2000工程力学习题精解	电气工程基础
统计学	工程力学	工程电磁场基础
离散数学	3000物理习题精解	数字信号处理
Mathematica使用指南	流体动力学	数字系统导论
数理金融引论	物理学基础	数字原理
机械振动	材料力学	电机与机电学
微分方程	2000离散数学习题精解	基本电路分析
统计学原理（上）	工程热力学	信号与系统
统计学原理（下）	数值分析	微生物学
微积分	量子力学	生物化学
静力学与材料力学	有机化学习题精解	生物学
有限元分析	3000化学习题精解	分子和细胞生物学
传热学	大学化学习题精解	人体解剖与生理学
近代物理学	电路	

http://www.sciencep.com http://www.mheducation.com

ISBN 7-03-009621-5



McGraw Hill

ISBN 7-03-009621-5/O・1516
定价：30.00元

全美经典学习指导系列

Mathematica 使用指南

〔美〕 D. 尤金 著

邓建松 彭冉冉 译

科学出版社

麦格劳-希尔教育出版集团

2002

内 容 简 介

本书是全美经典学习指导系列丛书之一的"Mathematica"的中文翻译版本，由著名大学具有丰富教学经验的教授编写，中国科学技术大学年轻的博士讲师翻译. 全书共分12章，全面介绍Mathematica语言在技术和科学计算中的广泛应用. 每章在介绍基本内容的基础上还精心设计了大量的例题和习题解答，以演示Mathematica软件的各种功能，例题和习题简练，切中要害. 读者无疑会为Mathematica的巨大功能所叹服，也能够应用这个软件解决各种数学难题.

本书适合高等院校本科生、研究生及科学技术人员使用.

Eugene Don: Schaum's Outline of Theory and Problems of Mathematica

ISBN: 0-07-135719-X

图字：01-2001-4493号

图书在版编目(CIP)数据

Mathematica使用指南/(美)D. 尤金著；邓建松，彭冉冉译. —北京：科学出版社，2002

(全美经典学习指导系列)

ISBN 7-03-009621-5

Ⅰ.M… Ⅱ.①尤…②邓…③彭… Ⅲ.数学-应用软件，Mathematica-指南 Ⅳ.O245-62

中国版本图书馆CIP数据核字(2001)第047880号

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

源海印刷厂 印刷

科学出版社总发行 各地新华书店经销

*

2002年1月第 一 版 开本：A4(890×1240)

2002年1月第一次印刷 印张：20

印数：1—4 000 字数：572 000

定价：30.00元

(如有印装质量问题，我社负责调换〈新欣〉)

前　言

本书是用来帮助学生以及在日常事务中需要用到数学知识的技术人员学习 Mathematica 的．计算机软件系统 Mathematica 可以进行复杂的数学运算．这里所采用的教学思路很简单，就是用示例来帮助学习．对最常用命令除了提供很容易看懂的描述外，还提供了 750 多个示例与习题，其每一个都经过精心设计，以演示 Mathematica 软件的各种功能．

本书介绍了在代数、三角、微积分、微分方程、线性代数中最常用到的命令和选项．许多例题和习题都很简练，切中要害．必要的地方还加上了说明，以便理解．

我们认为读者不应只是重复得到正文中给出的输出结果，还应该对代码进行一些修改，查看所做的修改对结果的影响．我认为这是学习这种独特程序的语法与能力的最有效方法．

本书前三章对 Mathematica 的语法和风格进行了介绍．后面几章的结构使得读者可以只在意自己所感兴趣的内容，并不必要全部通读．当然有时候会遇到在前面章节中介绍的命令，此时可以利用索引方便地找到对命令的描述．

无疑你会为 Mathematica 的巨大功能所叹服，我也衷心希望你能够应用这个软件的功能解决各种数学难题，尤其是那些在前几年还不可能解决的问题．

我也想借此机会感谢 McGraw-Hill 出版社的工作人员在本书写作过程中的帮助，特别要感谢 J．勒纳先生对本项目的鼓励与支持．

D．尤金

目　录

第一章　入　　门

1.1　记号与约定

实验表明,Mathematica 语言通过试验是很容易学习的,我们的建议就是读者应尽可能多地尝试示例和习题,并通过改变选项和参数进行尝试.事实上,可以认为本章是一个教程,帮助那些想马上使用 Mathematica 的读者着手使用这个软件.

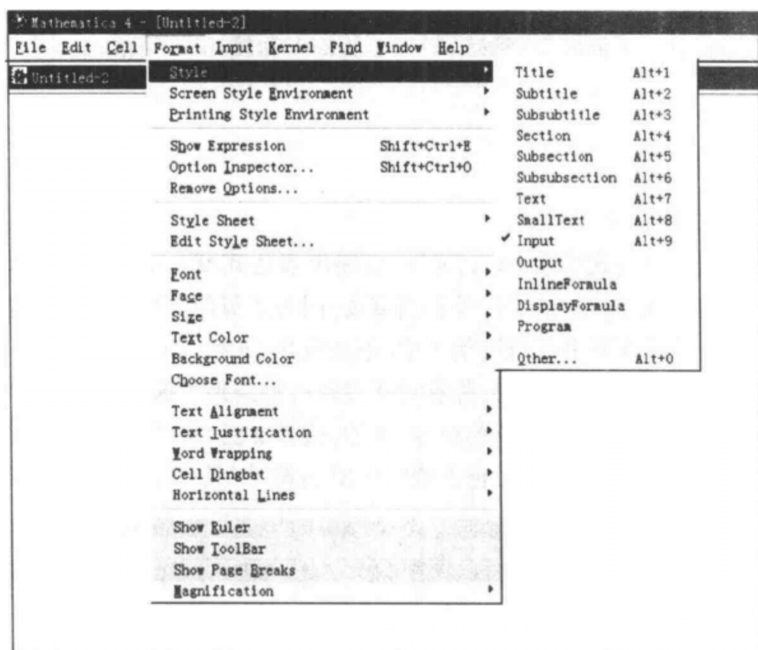
为了方便检索,我们在新介绍的命令前面加上■标签,在那些与该命令相应的选项前面加上 • 符号.

为了与 Mathematica 的约定一致,所有的指令都用粗体 Courier 字体表示,而 Mathematica 输出则用普通的 Courier 字体显示.如:

This line is written in Courier boldface type.

This line is wirtten in Courier regular face type.

在正文中用双箭头 (⇒) 表示菜单命令.例如,用 Arial 字体显示的 **Format ⇒Style ⇒Input** 就表示先到 **Format** 菜单,再到 **Style** 子菜单,然后在 **Input** 上点击.



在 Mathematica 中不时使用特殊记号 `,我们称它为反引号.不要把它与撇号混淆.

最后要指出的是,许多 Mathematica 命令使用箭头 (→) 表示命令内部的选项.如果愿意的话,可以用 - >(- 后接>)来代替.

本书中所有的例子都是在 Mathematica 版本 4 下执行的.如果用的是 Mathematica 版本 3, 那么要注意在计算机上的区别.最常见的差别有,在版本 3 中,输出结果中的自然对数的底显示为 E,而虚数单位显示为 I.与版本 3 不同,在版本 4 中则分别显示为e和i.在版本 4 中也会自动把 - >转化为→, 把 ! = 转化为≠.当然在版本 4 中还有其他的一些增强功能,但其中绝大多数对用户而言是很容易懂的.

1.2　内核与终端

内核就是 Mathematica 的计算引擎. 用户输入指令, 内核就给出反馈的结果, 结果可以是各种格式, 如数字、图形、矩阵或其他形式. 内核在后台无声地工作, 而且在大多数情形中, 用户是感觉不到的.

在用户与内核之间的界面就称为终端, 终端的主要组成部分是 Mathematica 的笔记本 (notebook). 通过笔记本, 用户不但可以与内核交流, 而且可以很方便地准备工作文件.

为了执行指令, 你需要先输入指令, 然后按 **[ENTER]** 键. 绝大多数微机键盘上都有两个 **[ENTER]** 键, 但只有位于键盘最右边的 **[ENTER]** 键才会执行指令. 另外一个 **[ENTER]** 键则需要与 **[SHIFT]** 键一起使用, 否则就只是得到新的一行. 如果正在使用的是 Macintosh 计算机, 那么注意不要把 **[ENTER]** 键与 **[RETURN]** 键混淆.

在下面例 1 中显示的图形就是标准的 Mathematica 界面. 在右边的符号构成 **BasicInput** (基本输入)模板, 通过这个模板, 只要点击鼠标, 就可以输入最常用的数学符号. (如果在屏幕上没有见到这个模板, 那么点击 **File ⇒Palettes ⇒BasicInput** 它就会出现.) 利用 **File ⇒ Palettes** 也可以得到其他的有专门用途的模板.

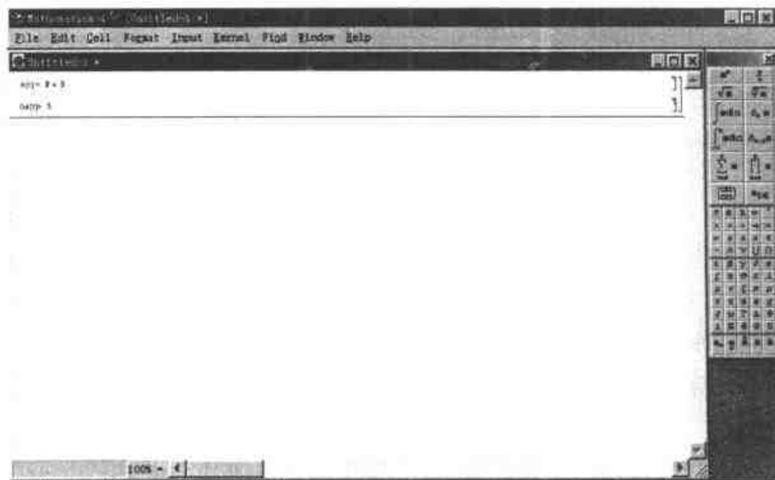
每个符号都可以在模板上点击得到. 如果使用的确实是模板, 那么这时笔记本就非常类似于数学教科书的页面. 本书中的许多例子都充分利用了 **BasicInput** 模板. 然而每个 Mathematica 符号都有另外一种描述性输入格式, 从而可以直接用键盘输入. 例如, π 可以用 **Pi** 表示, 而$\sqrt{5}$可以写为 **Sqrt[5]**. 对于那些倾向于不使用鼠标的 Mathematica 熟练用户而言, 这些表示是非常有用的.

在例 1 中的笔记本名称为 "Untitled-1", 它就是你输入命令的地方, 而且 Mathematica 也在这里显示计算的结果. 下面的图形显示了例 1 的输入和输出. (在 Macintosh 计算机上, 显示结果会稍有不同.)

例 1　计算 2 加上 3 的结果.

注意, 内核给输入表达式赋予 In[1]标签, 给输出表达式赋予 Out[1]标签. 这样用户就可以跟踪内核的运算指令的顺序. 这些标签相当重要, 因为运算的顺序并不一定与指令在笔记本上的实际位置一致. 然而在本书的所有例子中, 不会包含 In 和 Out 标签.

在尝试本书提供的例子与习题时, 你有可能发现所得结果与这里给出的不一样, 那么这就有可能是因为你已经定义了有指定值的符号. 例如, 假设 x 已经被定义为 3, 那么所有出现的 x 都要用 3 取代. 对于这种情况, 你应当首先清除所有的符号(见第 1.5 节)并再进行尝试. 本书



中所有的例子和习题都假设事先清除了那些符号.

在 Mathematica 的一次运行中,可以同时处理几个笔记本.然而,如果使用的只是一个内核,那么在一个笔记本内对符号的改变,将会对所有笔记本发生作用.

可能有时候你只需要执行表达式的一部分,那么可以先选择这部分要被处理的表达式,然而按[CTRL] + [SHIFT] + [ENTER]键(PC 机)或者[COMMAND] + [RETURN]键(Mac 机).

例 2 假设只希望对表达式 2 * 3 + 5 中的乘法部分进行处理,那么先选择 **2 * 3**:

```
2*3 + 5
```

然后按[CTRL] + [SHIFT] + [ENTER]键(PC 机)或者[COMMAND] + [RETURN]键(Mac 机)

```
6 + 5
```

有时候输入的指令可能需要计算很长时间,或者由于不小心造成无限循环,那么为了中断计算,可以使用菜单命令 **Kernel ⇒Abort Evaluation.** 另外,也可以按[ALT] + .(在 Macintosh 上为[COMMAND] + .)结束计算.如果这些措施不行的话,那么需要进入到 **Kernel ⇒Quit Kernel ⇒Local** 以结束内核.然而,这样做的话,就会失去所有的已定义符号和数值.但 Mathematica 笔记本并不会丢失,从而可以很容易地恢复失去的数据.

与所有的计算机软件相同,有时候 Mathematica 也会完全崩溃.那么这时解决问题的惟一方法就是关闭 Mathematica 并重新执行它,很少的情况不可能需要重新启动计算机.那么这时 Mathematica 笔记本中的内容都会丢失.因此经常**备份自己的笔记本**是相当重要的.

最后提一下,有时候你可能希望在 Mathematica 命令中包含注释.实际上所有包含在 *(* 和 *)之间的命令都会被内核所忽略.

例 3
```
12 + (* these words will be ignored by the kernel *) 3
15
```

习 题 解 答

1.1 12 乘以 17,再加上 9.

解 ☞

```
12 *17 + 9
213
```

1.2 在习题 **1.1** 中只进行 12 乘以 17 的运算,并不加上 9.

解 ☞

```
12*17 + 9            ←用鼠标选择 12 * 17.
```

按[CTRL] + [SHIFT] + [ENTER]键或者[COMMAND] + [RETURN]键(Mac)

```
204 + 9
```

1.3 下列程序是一个无限循环.执行这段程序,然后再中断运算.

$x = 1;$

While$[x > 0,\ x = x + 1]$

解 ☞

```
x = 1;
While[ x >0,x = x + 1]
```

█ + █

```
$ Aborted
```

1.4　17.2 乘以 16.3,然后再加上 4.7.

解 ☞

```
17.2 *16.3 + 4.7
285.06
```

1.5　17.2 乘以 16.3 与 4.7 的和

解 ☞

```
17.2 *(16.3 + 4.7)
361.2
```

1.6　计算 $2x+3$, $5x+9$, $4x+2$ 的和.

解 ☞

```
(2 x + 3) + (5 x + 9) + (4 x + 2)
14 + 11 x
```

1.3 Mathematica 的怪癖

Mathematica 区分字母的大小写

例如,**Integrate** 和 **integrate** 是不同的.所有的 Mathematica 命令都是以大写字母开头的.而其中有些命令(如 **FindRoot**)更使用多个大写字母.为了避免冲突,最好的主意就是所有的用户定义符号都用小写字母开头.

不同的括号有不同的用途

- 方括号用在函数参数指定中:**Sin[x]**(不是 **Sin(x)**).
- 圆括号表示分组.**(2 + 3) *4** 就表示先进行2 + 3,然后再乘以4.不要输入为**[2 + 3] *4.**
- 大括号表示列表:{**1, 2, 3, 4**}.关于列表的详情,见第三章.

自然对数的底是 **E** 而不是 **e**

既然每个 Mathematica 符号都以大写字母开头,自然对数的底又是 E,这很容易使人混淆,因此要非常小心这一点.同样,I(不是 i)表示虚数单位.如果愿意的话,可以使用模板中的e和i.

多项式并不是按"标准"形式显示的

Mathematica 在显示多项式时,把常数项放在最前面,然后从左到右按升幂排列.因此多项式 x^2+2x-3 显示为 $-3+2x+x^2$.

习题解答

1.7 利用 **Sqrt** 函数计算 $\sqrt{81}$. 如果没有使用大写字母 S 会出现什么情况?

解 ☞

```
Sqrt[81]
9
sqrt[81]
General::spell : Possible spelling error : new symbol name "sqrt" is similar to
  existing symbol "Sqrt".
```

(可能拼写出错:新符号名称 "sqrt" 类似于已有符号 "Sqrt".)

```
Sqrt[81]
```

1.8 利用括号计算 2 与 3 的和乘以 5 与 7 的和. 如果使用方括号会发生什么情况?

解 ☞

```
(2 + 3)(5 + 7)
60
[2 + 3][5 + 7]
Syntax::txntxi : "[2 + 3]" is incomplete; more input is needed.
```

("[2 + 3]" 为不完整表达式,需要更多的输入.)

```
[2 + 3][5 + 7]
```

1.9 利用 **Sin** 函数计算 sin(π/2). 如果使用的是圆括号,会发生什么情况?

解 ☞

Sin[Pi/2] 或 **Sin[π/2]**

1

Sin(Pi/2)

$\frac{\pi \sin}{2}$

Mathematica 认为你希望把符号 **sin** 乘以 π, 然后再除以 2.

1.10 张三在 Mathematica 运行时输入了 **[4 + 1] * [6 + 2]**, 那么他为什么没有得到答案 40?

解 ☞

方括号不能用于表示分组,这时必须用圆括号.

1.11 当李四输入 **sqrt[9]** 时,他为什么没有得到答案 3?

解 ☞

Mathematica 函数必须以大写字母开头.

1.12 当王五输入 **Cos(0)** 时,他为什么没有得到答案 0?

解 ☞

包围函数参数值的括号是方括号,而不是圆括号.

1.4 Mathematica 给出精确解答

Mathematica 软件经过了精心设计，可以如同一位数学家那样工作，也就是非常讲究精确性. 因此在这里你不会得到类似于计算器上那样的 10 位或 12 位近似数字，而是一个 100% 准确的数学表达式.

例 4 $\sqrt{12}$

$2\sqrt{3}$

例 5 1/3 + 3/5 - 5/7 + 2/11

$\frac{463}{1155}$

例 6 $\pi+\pi$

2π

例 7 $\sqrt{-1}$

i

习题解答

1.13 化简$\sqrt{2}+\sqrt{8}+\sqrt{18}$.

解 ☞

$\sqrt{2}+\sqrt{8}+\sqrt{18}$ 或者 Sqrt[2] + Sqrt[8] + Sqrt[18]

$6\sqrt{2}$

1.14 计算 3, 5, 7, 11 和 13 的倒数和.

解 ☞

$\frac{1}{3}+\frac{1}{5}+\frac{1}{7}+\frac{1}{11}+\frac{1}{13}$ 或者 1/3 + 1/5 + 1/7 + 1/11 + 1/13

$\frac{12673}{15015}$

1.15 利用 Sqrt 函数准确计算 π 的平方根.

解 ☞

Sqrt[Pi]

$\sqrt{\pi}$ ←这是准确表示 π 的平方根的惟一方法.

1.16 $\sqrt{8}$乘以$\sqrt{2}$.

解 ☞

$\sqrt{8}\sqrt{2}$或者 Sqrt[8] * Sqrt[2]

4

1.17 化简$\sqrt{3}+\sqrt{12}+\sqrt{27}+\sqrt{48}$，保留答案中的根号形式.

解 ☞

$\sqrt{3}+\sqrt{12}+\sqrt{27}+\sqrt{48}$

$10\sqrt{3}$

1.5　Mathematica 基础

本节讨论 Mathematica 中的几个简单概念. 在后续章节中将会对每个概念详加解释.

符号定义为一串字母、数字和几个特定字符构成的序列, 但不能以数字开头. 一旦定义了符号, 那么只有对符号进行了修改或者清除, 才会改变它的值.

算术运算是利用 +, -, *, / 符号进行的显式操作. 指数用 ^ 符号表示, 因此 **x ^y** 就表示 x^y. 与代数中约定一样, 两个符号之间没有运算符意味着乘法, 因此 **2a** 与 **2 * a** 是相同的. 然而如果是两个符号相乘必须小心处理, 因为 **ab** 表示单个符号, 这个符号以 **a** 开头, **b** 结尾. 为了表示 **a** 乘以 **b**, 那就必须在两个字母之间加入 * 或 ((在 **BasicInput** 模板上)或者空格, 即输入格式为 **a * b**, **a × b** 或 **a b**.

例 8　**a = 2**

2

b = 3

3

c = a + b

5

注意每次运算后显示的结果. 有时候这种显示方式令人心烦, 那么可以在指令的右边加上分号(;)去掉它.

例 9　**a = 2;**

b = 3;

c = a + b

5

运算的执行顺序为: (a) 指数, (b) 乘法与除法, (c) 加法与减法. 如果要改变运算的顺序, 那么就必须用圆括号 ( ). 注意不能用方括号[ ] 或大括号{ }达到这个目的.

例 10　**2 + 3 *5**

17

(2 + 3) *5

25

在 Mathematica 中的每个符号都表示一定的内容. 它既有可能是某个简单的数值运算结果, 也有可能是一个复杂的代数表达式.

例 11　**a = 3 ;**

$\mathbf{b} = \sqrt{\dfrac{\mathbf{x}^2+\mathbf{1}}{\mathbf{2x+3}}}$**;**

这里 a 就是表示数值 3 的一个符号，而 b 是表示非有理代数表达式的一个符号.

如果你忘记了符号所代表的意义，那么只要输入 ? 后接符号的名称就可以知道符号的定义了.

例 12 ? a

Global`a

a = 3

? b

Global`b

b = $\sqrt{\dfrac{1+x^2}{3+2x}}$

为了删除一个符号，以便使得它可用于其他的目的，那么可以用 **Clear** 或 **Remove** 命令.

■ **Clear[符号名]** 清除指定符号的定义及取值，但并没有清除它的属性、信息或默认值，因此指定的符号仍在 Mathematica 的符号清单中.

■ **Remove[符号名]** 完全删除指定的符号. 因此除非重新进行了定义，否则不再识别这个符号.

符号名 = . 也会删除指定符号的定义.

例 13 **Clear[a]**

? a ←? a 给出符号 a 的信息.

Global`a

Remove[b]

? b

Information::notfound : Symbol b not found.（没有找到符号 b.）

N 命令可以计算近似值.

■ **N[表达式]** 给出表达式的 6 位有效数字（Mathematica 的默认值）的近似值.

■ **N[表达式, n]** 尽力给出表达式的达 n 位有效数字的近似值.

进行近似计算的另外一种方便的方法是在被近似表达式右边用 **//N**. 这等价于 **N[表达式]**. // 也可以用于其他的 Mathematica 命令.

■ **"表达式//命令"** 等价于 **"命令[表达式]"**.

例 14 **N[π]** 或者 **π //N**

3.14159

N[π, 50]

3.1415926535897932384626433832795028841971693993751

在 Mathematica 核中可以跟踪前面的计算结果. 符号 **%** 返回前一次计算的结果，而 **%%** 返回再前一次的计算结果，**%%%** 依此类推. 灵活地使用 **%** 符号可以节省大量的输入时间.

例 15 为了构造 $\sqrt{\pi+\sqrt{\pi+\sqrt{\pi}}}$，我们输入 **Sqrt[Pi + Sqrt[Pi + Sqrt[Pi]]]**. 而另外一种可能不容易使人糊涂的输入方法为

---

① 除非在表达式中的数值为精确的或者有足够高的精度，Mathematica 可能无法返回 n 位有效数字的近似值

```
Sqrt[Pi];                  ←分号禁止显示中间计算的输出.
Sqrt[Pi + %];
Sqrt[Pi + %]
```

$\sqrt{\pi+\sqrt{\pi+\sqrt{\pi}}}$

而下面是利用 **BasicInput** 模板的输入方法：

$\sqrt{\pi}$;

$\sqrt{\pi+\%}$;

$\sqrt{\pi+\%}$

$\sqrt{\pi+\sqrt{\sqrt{\pi}+\pi}}$

习 题 解 答

1.18 定义 $a = 3$, $b = 4$, $c = 5$. 然后用 a 与 b 的和乘以 b 与 c 的和，只显示最后的结果.

解 ☞

```
a = 3;
b = 4;
c = 5;
(a + b) * (b + c)
63
```

1.19 令 $a = 1$, $b = 2$, $c = 3$，然后把 a, b, c 加在一起. 接着从内核的存贮空间中清除 a, b, c，再计算它们的和.

解 ☞

```
a = 1;
b = 2;
c = 3;
a + b + c
6
Clear[a, b, c]
a + b + c
a + b + c
```

1.20 给出自然对数基底 e 的有 25 位小数的近似值.

解 ☞

```
N[E, 26] 或者 N[e, 26]          ←这里需要26位有效数字以给出25位小数.
2.7182818284590452353602875
```

1.21 (a) 把 $\frac{1}{7}+\frac{2}{13}-\frac{3}{19}+\frac{1}{23}$ 表示为单个分数.

(b) 给出上述表达式的有 20 位小数的近似值.

解 ☞

```
1/7 + 2/13 - 3/19 + 1/23
```

$\frac{7249}{39767}$

```
N[%, 20]
```

0.18228682073075665753

1.22 按下列要求计算 $\sqrt{968}$的结果:(a) 精确;(b) 近似到25位有效数字.

解 ☞

$\sqrt{968}$或者 `Sqrt[968]`

$22\sqrt{2}$

```
N[%, 25]
```

31.11269837220809107363715

1.23 计算12乘以6的结果.然后再计算15乘以7的结果.接着用 % 和 %% 把两个结果加在一起.

解 ☞

```
12 *6
```

72

```
15 *7
```

105

```
% + %%
```

177

1.24 计算 $1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{2}}}}$.

解 ☞

$1+\frac{1}{2}$

$\frac{3}{2}$

$1+\frac{1}{\%}$

$\frac{5}{3}$

$1+\frac{1}{\%}$

$\frac{8}{5}$

$1+\frac{1}{\%}$

$\frac{13}{8}$

1.25 计算 $1+(1+(1+(1+(1+1)^2)^2)^2)^2$ 的值.

解 ☞

```
1 + 1
```

```
2
1 + % ^2
5
1 + % ^2
26
1 + % ^2
677
1 + % ^2
458330
```

1.6 单元

单元是 Mathematica 笔记本的构造模块，在笔记本右边用蓝色的方括号表示每个单元.(很可能你早就注意到这些方括号，并奇怪它们到底是做什么用的.)单元中可以包含子单元，并进而包含子子单元，依此类推.

内核在计算笔记本时是以逐个单元为基础的，因此如果你把几条指令放在同一个单元中，那么只要按一下**ENTER**键，它们就都会被执行.

例 16

```
a = 1 + 2
b = 2 + 7      ←这三行包含在同一个单元中，因此只需要按一下ENTER键.
c = a + b
3
9
12
```

通过移动鼠标，直到指针变为一条水平线时单击，就可以形成一个新的单元.这时将会在屏幕上出现一条贯穿的水平线，标志新单元的开始.可以利用 **Cell⇒Divide Cell**(部分单元)命令剖分单元，这样单元就会被分成两个单元，断点出现在当时指针所处的位置.

为了组合几个单元，可以先选择相应单元的方括号(这时会出现一条竖线)，然后点击 **Cell⇒Merge Cell**(合并单元).这些操作都有键盘快捷键，就列在相应菜单项的后面.

为了避免太长的笔记本，双击单元的括号就可以关闭(或压缩)单元.这时括号的外观发生改变，就好像一个鱼钩.再次双击其标志就可以打开单元.

为了满足各种用途的需要，在 Mathematica 中有不同类型的单元.只有输入单元可以送到内核进行运算.文本单元用于进行描述性说明.而其他类型的单元(如标题、子标题、节、小节等)的创建可以利用菜单 **Format ⇒Style** 做到.如果愿意的话，点击 Format⇒Show Toolbar(显示工具条)可以显示一个方便的工具条.这样就会在笔记本窗口的顶部显示单元的类型.

习 题 解 答

1.26 令 $a = 2x + 3, b = 5x + 6$，然后计算 $a + b$.

(a) 把每个指令单独放在一个单元中，并分别执行.

(b) 把所有三条指令放在同一个单元中，并同时执行.

解 ☞

下面就是执行后所得结果的样子.

(a)

```
a=2x+3
3+2x
b=5x+6
6+5x
a+b
9+7x
```

(b)

```
a=2x+3
b=5x+6
a+b
3+2x
6+5x
9+7x
```

1.27 令 $a = 2x + 3y + 4z, b = x + 3y + 5z, c = 3x + y + z$,并计算 a, b, c 的和.把四行放在同一个单元中,并执行它.只显示最后的结果.

解 ☞

```
a=2x+3y+4z;
b=x+3y+5z;
c=3x+y+z;
a+b+c

6x+7y+10z
```

1.7 寻求帮助

在 Mathematica 中可以从许多地方得到帮助.第一个地方,也是最重要的一个地方,就是 Mathematica 手册.第二个地方,也可能是最方便的一个地方,就是内置在软件中的帮助索引.在帮助索引中你会发现所有的命令分组排列,只要点击相应条目,就可以得到所需要的帮助信息.实际上整本 Mathematica 手册也内置在这个软件中.

在帮助文件中有无数个例子,你可以执行它们.在帮助文件中可以任意进行修改,不用担心改变了它的内容.因为这些文件是处于被保护模式,所做的修改不会永久被保存下来.

如果已知所用命令的名称,那么可以用问号 ? 后接命令名称来确定它的语法.关于该命令的更多信息,包含它的属性和选择,可以用 ?? 得到.

有时候,当你不小心出错时,Mathematica 就会叫一声.如果你不知道为什么会这样,那么只要到 **Help ⇒Why The Beep**? 中就会得到提示.

例 17 假设你知道 **Plot** 命令能绘制函数的图像,但没记住它的语法,那么可以如下操作:

?

Plot

Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to xmax. Plot[{f1, f2,…}, {x, xmin, xmax}] plots several functions fi.

(Plot[f, {x, xmin, xmax}]生成函数 f 的图形,其自变量为 x,绘图区间为[xmin, xmax].Plot[{f1, f2, … }, {x, xmin, xmax}]绘制几个函数 fi 的图形.)

如果需要知道关于属性与选项设置(以及它们的默认值)的信息,那么可以用 ??.

?? **Plot**

Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to xmax. Plot[{f1, f2, …}, {x, xmin, xmax}] plots several functions fi.

(Plot[f, {x, xmin, xmax}]生成函数 f 的图形,其自变量为 x,绘图区间为[xmin, xmax].Plot[{f1, f2, …}, {x, xmin, xmax}]绘制几个函数 fi 的图形.)

```
Attributes[Plot] = {HoldAll, Protected}

Options[Plot] = {AspectRatio → 1/GoldenRatio, Axes → Automatic,
 AxesLabel → None, AxesOrigin → Automatic,
 AxesStyle → Automatic, Background → Automatic,
 ColorOutput → Automatic, Compiled → True,
 DefaultColor → Automatic, Epilog → {}, Frame → False,
 FrameLabel → None, FrameStyle → Automatic,
 FrameTicks → Automatic, GridLines → None,
 ImageSize → Automatic, MaxBend → 10., PlotDivision → 30.,
 PlotLabel → None, PlotPoints → 25, PlotRange → Automatic,
 PlotRegion → Automatic, PlotStyle → Automatic,
 Prolog → {}, RotateLabel → True, Ticks → Automatic,
 DefaultFont :→ $DefaultFont,
 DisplayFunction :→ $DisplayFunction,
 FormatType :→ $FormatType, TextStyle :→ $TextStyle}
```

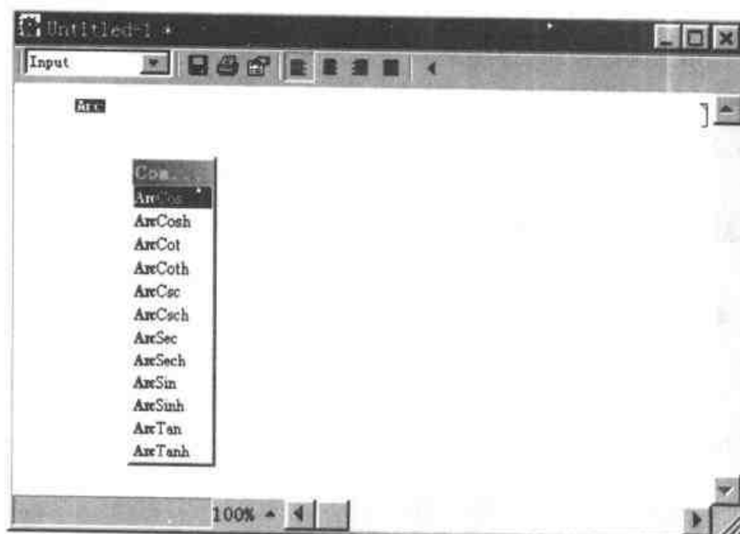
也可以用 **Options** 命令得到选项信息.当你想知道某个特定选项,却记不住它的名称时这条命令是相当有用的.

例 18 **Options[Solve]**

```
{InverseFunctions → Automatic, MakeRules → False, Method → 3,
Mode → Generic, Sort → True, VerifySolutions → Automatic,
WorkingPrecision → ∞}
```

更经常遇到的情况是你只记住了符号名称的一部分,而不知道它的完整名称.例如,若你知道符号名称开头为 Arc,那么就输入你知道的那部分,并按 CTRL + K.这样就会生成由所有的以 Arc 开头的命令和函数组成的菜单.在你需要的名称上点击就可以了.如果你用的是 Macintosh 计算机,那么就用 COMMAND + K.(COMMAND 键就是上面有一个苹果标志的按键.)

例 19 输入 Arc 并按 CTRL + K 或者 COMMAND + K.



另外一种确定符号名称的方法是用 ? 外加通配符. 字符 * 就是通配符, 可以代替任何字符序列. 通配符可以用在任何地方, 既可以用在符号的开头, 中间, 也可以用在结尾.

例 20 (a) 找出所有以 'Inv' 开头的命令.

```
? Inv *
Inverse                          InverseJacobiDC
InverseBetaRegularized           InverseJacobiDN
InverseEllipticNomeQ             InverseJacobiDS
InverseErf                       InverseJacobiNC
InverseErfc                      InverseJacobiND
InverseFourier                   InverseJacobiNS
InverseFunction                  InverseJacobiSC
InverseFunctions                 InverseJacobiSD
InverseGammaRegularized          InverseJacobiSN
InverseJacobiCD                  InverseSeries
InverseJacobiCN                  InverseWeierstrassP
InverseJacobiCS
```

(b) 找出所有以 'in' 结尾的命令.

```
? * in
ArcSin              GroupPageBreakWithin    Sin
AxesOrigin          Join                    StackBegin
Begin               Min                     StringJoin
CoefficientDomain   PageBreakWithin         $MachineDomain
ConstrainedMin      Plain
```

(c) 找出所有在中间包含 'our' 的命令.

```
? * our *
ButtonSource        ContourShading          InverseFourier
ContourGraphics     ContourSmoothing        LightSources
ContourLines        ContourStyle            ListContourPlot
ContourPlot         Fourier                 MovieContourPlot
Contours
```

通配符也可以用来确定到现在为止内核中已用到了哪些符号. 输入 ?` * 就返回在此次执行 Mathematica 时已定义符号的清单. 反引号字符 ` 表示全局性, 即得到的是所有全局符号的清单. (见附录 A.2.)

例 21 **注意** 这个例子的结果可能与你所用计算机上显示的稍有不同.

```
a = 3;
b2xy = 4;
c = 5;
?` *
a        b2xy        xyz7
```

Clear["`*"]** 会清除所有全局符号. **Remove["` *"] 则会删除所有全局符号.

例 22

```
Remove["` *"]
?` *                    ← 检查是否还有任何符号.
```

```
Information::nomatch : No symbol matching` * found.
```
(没有找到与` * 匹配的符号.)

习题解答

1.28 查找 Mathematica 命令 **Simplify** 的基本信息.

解 ☞

```
? Simplify
Simplify[expr] performs a sequence of algebraic transformations on expr, and
returns the simplest form it finds.
```
(Simplify[expr]对 expr 进行一系列的代数变换,并返回它得到的最简形式.)

1.29 查找 Mathematica 命令 **Simplify** 的包含选项默认值的扩展信息.

解 ☞

```
?? Simplify
Simplify[expr] performs a sequence of algebraic transformations on expr, and
 returns the simplest form it finds.
```
(Simplify[expr]对 expr 进行一系列的代数变换,并返回它得到的最简形式.)
```
Attributes[Simplify] = {Protected}
Options[Simplify] = {ComplexityFunction → Automatic, TimeConstraint → 300,
Trig → True}
```

1.30 查找 Mathematica 命令 **Factor** 的帮助信息,并利用它分解因式:$x^3-6x^2+11x-6$.

解 ☞

```
? Factor
Factor[poly] factors a polynomial over the integers.
 Factor[poly, Modulus→p] factors a polynomial modulo a
 prime p. Factor[poly, Extension → {a1, a2, …}] factors a
 polynomial allowing coefficients that are rational
 combinations of the algebraic numbers ai.
```
(Factor[poly]在整数集上分解多项式.Factor[poly, Modulus→p]对模素数 p 进行分解多项式.Factor[poly, Extension→{a1, a2,… }]在允许系数为代数数 ai 的有理组合的条件下分解多项式.)

```
Factor [x^3 - 6x^2 + 11x - 6 ]
(-3 + x)(-2 + x)(-1 + x)
```

1.31 找出所有以 Abs 开头的 Mathematica 命令.

解 ☞

```
? Abs *
Abs                          AbsoluteThickness
AbsoluteDashing              AbsoluteTime
AbsolutePointSize
```

1.32 找出所有以 Sin 开头，al 结尾的 Mathematica 命令.

解 ☞

```
? Sin * al
SinhIntegral                 SinIntegral
```

1.33 找出所有以 Fi 开头的 Mathematica 命令.

解 ☞

```
? Fi *
Fibonacci                    Find
File                         FindList
FileBrowse                   FindMinimum
FileByteCount                FindRoot
FileDate                     First
FileInformation              Fit
FileName                     FixedPoint
FileNames                    FixedPointList
FileType
```

1.34 找出所有以 Fi 开头，t 结尾的 Mathematica 命令.

解 ☞

```
? Fi * t
FileByteCount                Fit
FindList                     FixedPoint
FindRoot                     FixedPointList
Fit
```

1.8 软件包

当 Mathematica 刚被执行时，有许多很专业化的函数和程序并没有被上载进系统. 如果需要它们的话，那就必须单独从 Mathematica 的硬盘目录中上载. 这些文件的形式为文件名.m.

例 23 半径为 1 的圆的隐式方程为 $x^2+y^2=1$. 为了绘制隐式方程确定的图像，需要使用命令 **ImplicitPlot**. 然而，只有在软件包 **Graphics** 中 **ImplicitPlot** 命令才可以使用. (相关信息可以到帮助索引下的 add-ons(其他)中查到.)为了上载这件图形命令，只需要输入

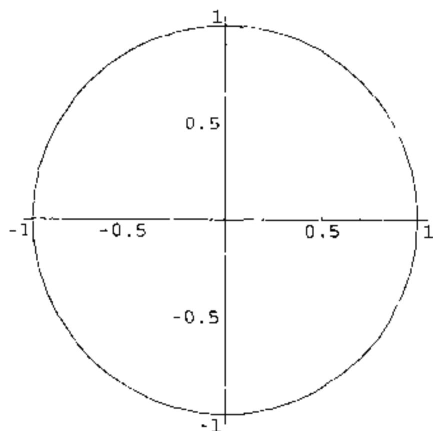
```
<<Graphics`ImplicitPlot`
```

如果所用计算机有足够的内存，那么可以输入**<<Graphics`**上载所有的图形软件包，这样 Mathematica 就会在调用其中每条图形命令时，自动上载相应部分. 现在就可以调用适当的图形命令了.

```
ImplicitPlot[x^2 + y^2 == 1, {x, -1, 1}];
```

注意 在 ImplicitPlot 命令中用的是两个等号. 我们将在第四章(二维图形)中详细讨论 ImplicitPlot 命令.

一旦上载了一个软件包，那么可以用 **Names** 命令得到这个软件包中包含的所有函数名称的清单.



例 24 **≪Miscellaneous\`Calendar\`** ←上载软件包 Miscellaneous\`Calendar\`.

Names["Miscellaneous\`Calendar\`*"]

{"Calendar","CalendarChange", "DayOfWeek", "DaysBetween",
"DaysPlus","EasterSunday","EasterSundayGreekOrthodox",
"Friday", "Gregorian", "Islamic", "JewishNewYear", "Julian",
"Monday", "Saturday", "Sunday", "Thursday", "Tuesday",
"Wednesday"}

例 25 在应用软件包时经常出现一种错误,这就是还没有上载相应的软件包,就使用了该软件包中的命令.例如,在上载图形软件包之间就执行 **ImplicitPlot**,会得到如下结果:

ImplicitPlot[x^2 + y^2 == 1, {x, -1, 1}]

ImplicitPlot[x^2 + y^2 == 1, {x, -1, 1}]

如果现在上载图形软件包,那么得到如下出错信息.

≪Graphics\`ImplicitPlot\`

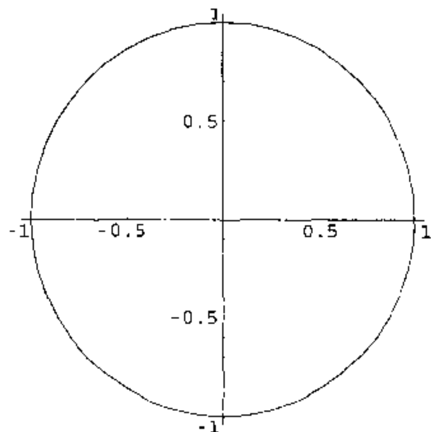
ImplicitPlot::shdw:

Symbol ImplicitPlot appears in multiple contexts ≪1≫;

Definitions in context Graphics\`ImplicitPlot\` may Shadow or be shadowed by other definitions.

(符号 ImplicitPlot 出现在多个上下文≪1≫中;在上下文 Graphics\`Implicit-Plot\`中的定义可能会覆盖其他定义,或者被其他定义覆盖.)

这时输入画图命令,并不能得到结果,因为现在有两个 ImplicitPlot 符号,这就是由在上载软件包之前的错误调用造成的.解决这个问题的方法就是从符号表中去掉该命



令,然后再画图:

```
Remove[ImplicitPlot]
ImplicitPlot[x^2 + y^2 == 1, {x, -1, 1}];
```

习题解答

1.35 在软件包 **DiscreteMath`Combinatorica`** 中包含了与组合数学和图论有关的多达 230 个函数.其中有一个函数叫 **KSubsets**,它列出给定集合的所有大小为 k 的子集.上载这个软件包,并执行**KSubsets[{1,2,3,4,5}, 3]**.这就会列出 {1, 2, 3, 4, 5} 的所有包含三个元素的子集.

```
<<DiscreteMath`Combinatorica`
KSubsets[{1,2,3,4,5}, 3]
{{1, 2, 3}, {1, 2, 4}, {1, 2, 5}, {1, 3, 4}, {1, 3, 5},
{1, 4, 5}, {2, 3, 4}, {2, 3, 5}, {2, 4, 5}, {3, 4, 5}}
```

1.36 函数 **DayOfWeek** 位于软件包 **Miscellaneous`Calendar`** 中,它给出日历中任一日期到底是星期几.上载这个软件包,查找帮助信息,从而获取该命令的语法,然后确定 2000 年 1 月 1 日是星期几.

```
<<Miscellaneous`Calendar`
? DayOfWeek
DayOfWeek[{y, m, d}, cal] gives the day of the week for year y, month m, and day
  d in calendar cal. The default calendar is the usual American calendar. The
  date can also
be given in {y, m, d, h, m, s} form.
```

(DayOfWeek[{y, m, d}, cal]算出在日历 cal 中 y 年 m 月 d 天是星期几.默认日历就是通常的美国日历.日期也可以按{y, m, d, h, m, s}形式给出.)

```
DayOfWeek[2000, 1, 1]
Saturday
```

1.37 软件包 **Miscellaneous`ChemicalElements`** 中包含的函数可以提供元素的化学与物理属性方面的重要信息.这个软件包中有两个函数,名称分别为 **AtomicWeight** 和 **AtomicNumber**,其定义从名称上看就很清楚了.用这两个函数计算钛(Titanium)的原子量和原子序号.

解 ☞

```
<<Miscellaneous`ChemicalElements`
? Titanium
Titanium is a chemical element.
```

(钛为一种化学元素.)

```
? AtomicWeight
AtomicWeight[element] gives the atomic weight of the specified element.
```

(AtomicWeight[element]给出指定元素的原子量.)

```
? AtomicNumber
AtomicNumber[element] gives the atomic number of the specified element.
```

(AtomicNumber[element]给出指定元素的原子序数.)

```
AtomicWeight[Titanium]
```

47.867

AtomicNumber[Titanium]

22

1.9 后续章节内容预览

如果你刚刚购买了 Mathematica 软件,那么就会迫不及待地想测试一下它的功能.下面给出的就是可供测试用的问题,紧接着给出基本的命令.为了简化,这里忽略了选项,从而使用的是 Mathematica 默认值.我们将会在后续章节中讨论如何改变这些命令,而现在只管放心品尝它们吧!

习题解答

1.38 给出$\sqrt{\pi}$有 50 位有效数字的近似值.

解 ☞

N[$\sqrt{\pi}$, **50]** 或者 **N[Sqrt[Pi], 50]**

1.7724538509055160272981674833411451827975494561224

注意这个近似值只有 49 位小数.

1.39 求解代数方程 $x^3 - 2x + 1 = 0$.

解 ☞

Solve[x^3 **- 2x + 1 == 0]** 或者 **Solve[x^3 - 2x + 1 == 0]**

$$\left\{\{x \to 1\}, \left\{x \to \frac{1}{2}(-1-\sqrt{5})\right\}, \left\{x \to \frac{1}{2}(-1+\sqrt{5})\right\}\right\}$$

1.40 把 $(1+x)^{10}$展开成多项式形式.

解 ☞

Expand[$(1+x)^{10}$**]** 或者 **Expand[(1 + x)^10]**

$1 + 10x + 45x^2 + 120x^3 + 210x^4 + 252x^5 + 210x^6 + 120x^7 + 45x^8 + 10x^9 + x^{10}$

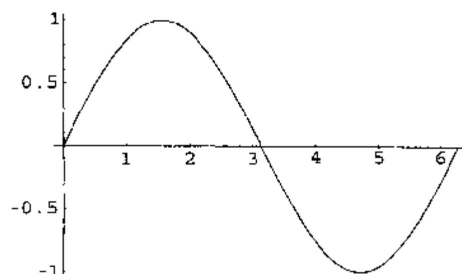
1.41 第 1000 个素数是什么?

解 ☞

Prime[1000]

7919

1.42 画出函数 $y = \sin x$ 从 0 到 2π 的图像.



解 ☞

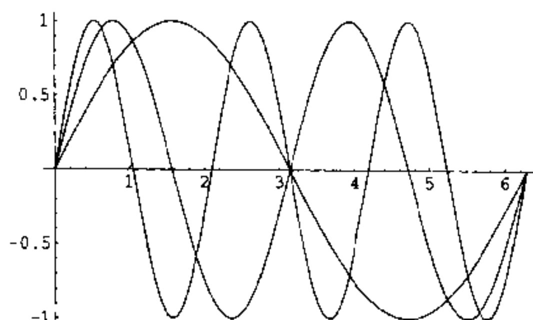
`Plot[Sin[x],{x, 0, 2 π}];` 或者 `Plot[Sin[x], {x, 0, 2 * Pi}];`

1.43 在同一坐标系中，画出函数 $y = \sin x$, $y = \sin 2x$ 与 $y = \sin 3x$, $0 \leqslant x \leqslant 2\pi$ 的图像.

解 ☞

`Plot[{Sin[x], Sin[2x], Sin[3x]}, {x, 0, 2π}];` 或者

`Plot[{Sin[x], Sin[2x], Sin[3x]}, {x, 0, 2 * Pi}];`

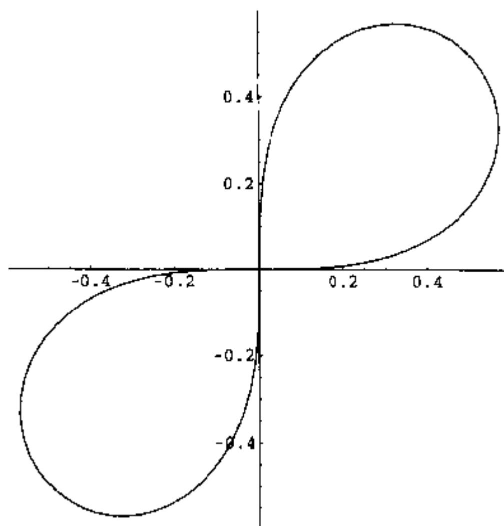


1.44 绘制 $(x^2 + y^2)^2 = xy$, $-2 \leqslant x \leqslant 2$ 的图像.

解 ☞

因为这是一个隐式方程，必须在 **Graphics** 软件包中首先输入 **Implicitplot** 命令。

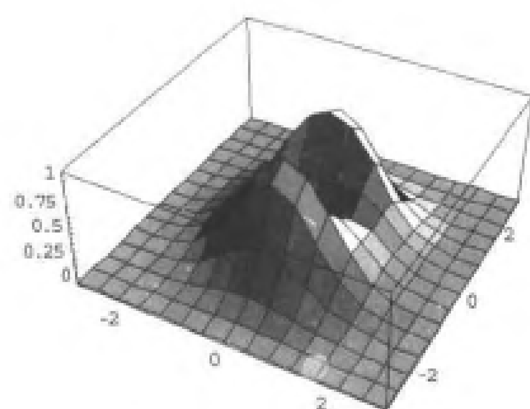
```
<<Graphics`ImplicitPlot`
ImplicitPlot[(x² + y²)² == x * y, {x, - 2, 2}]; 或者
ImplicitPlot[(x ^ 2 + y ^ 2) ^ 2 == x * y, {x, - 2, 2}];
```



1.45 绘制由 $z = (x^2 + 3y^2)e^{-(x^2+y^2)}$ 定义的三维空间中的曲面.

解 ☞

```
Plot3D[(x² + 3y²)E - (x² + y²), {x, - 3, 3}, {y, - 3, 3}]; 或者
Plot3D[(x ^ 2 + 3y ^ 2) * Exp[ - (x ^ 2 + y ^ 2)], {x, - 3, 3}, {y, - 3, 3}];
```


1
0.75
0.5
0.25
0
-2
0
2
2
0
-2

第二章　基本概念

2.1　常数

在 Mathematica 中使用预定义的符号表示内置数学常数.

- **Pi** 或 π 表示圆的周长与直径的比率.
- **E** 或 e 表示自然对数的底.

注意对 **Pi** 和 **E** 的处理都是符号性的,而不采用具体的数值,但却可以近似到任意精度.

例 1　N[π, 50] 计算出 π 的达 50 位有效数字(49 位小数)的近似值.

```
N[π, 50]
3.1415926535897932384626433832795028841971693993751
```

- **Degree** 等于 **Pi/180**,通常用来把角度转化为弧度.
- **GoldenRatio** 的值为 $(1+\sqrt{5})/2$,它对 Fibonacci 数列的处理是非常重要的.在 Mathematica 中也用它作为显示二维图形时默认的宽度与高度比率.
- **Infinity** 或∞是一个有特殊性质的常数.例如,$\infty+1=\infty$.
- **EulerGamma** 就是欧拉(Euler)常数,近似值为 0.577216.在积分和渐近展开中要用到它.
- **Catalan** 就是 Catalan 常数,近似值为 0.915966.在组合函数论中要用到它.

例 2　∞ + ∞是多少?

```
∞ + ∞
∞
```

习题解答

2.1　90 度近似等于多少弧度?

解 ☞

```
90 Degree //N
1.5708
```

←表达式//N 与 N[表达式]是等价的.

2.2　证明 **GoldenRatio** 满足代数方程 $x^2-x-1=0$.

解 ☞

```
x = GoldenRatio;
x^2 - x - 1 //N
0
```

2.3　如果从∞中减去∞会出现什么情况?

解 ☞

```
∞ - ∞
∞::indet :
```

```
Indeterminate expression −∞ + ∞ encountered.
```

(遇到了不定表达式 −∞ + ∞.)

```
Indeterminate
```

2.4 计算出自然对数底 e 的有 20 位小数的近似值.

解 ☞

```
N[E, 21]
2.71828182845904523536
```

2.2 内置函数

本节讨论由 Mathematica 所提供的一些比较常用的函数.由于可用的函数数目实在庞大,因此我们不会给出所有函数的用法.在后续章节中讨论了其他一些函数.

如果要调用标准数学函数,那么可以直接输入它的名称,也可以在 Mathematica 模板中点击它的符号.例如,为了计算一个数的平方根,可以使用函数 **Sqrt**,也可以使用 **BasicInput** 模板中的 $\sqrt{\ }$ 符号.注意函数的参数值必须放在方括号 [ ]内.

■ **Sqrt[x]** 或 $\sqrt{x}$ 给出 x 的算术平方根.

例 3 **Sqrt[1521]** 或者 $\sqrt{1521}$

39

高阶方根可以利用 $\sqrt[n]{x} = x^{1/n}$ 来计算.在 **BasicInput** 模板上的 $\sqrt[\square]{\ }$ 符号也可以实现同样的功能.注意负数的高阶方根是用特殊的格式给出的.

例 4 8 的三次方根是直接给出的,而 −8 的三次方根则是用 $\sqrt[3]{-1}$ 表示的.

8^(1/3) 或者 $\sqrt[3]{8}$

2

(−8)^(1/3) 或者 $\sqrt[3]{-8}$

$2(-1)^{1/3}$

例 5 **N[**$\sqrt{2}$**]**

1.41421

N[$\sqrt{2}$**, 50]**

1.4142135623730950488016887242096980785696718753769

Abs 函数给出 x 的绝对值 $|x|$.

■ **Abs[x]** 如果 x≥0,返回 x;如果 x<0,返回 −x.

Abs 函数也可以作用到复数上.如果 z 为复数 x + yi,那么 Abs[z]返回它的模,即 $\sqrt{x^2+y^2}$

例 6 **Abs[5]**

5

Abs[−5]

5

Abs[5 + 12I]

13

能确定一个数的符号的函数在有些问题中是有用的.

■ **Sign[x]**函数根据 x 为负数、零或者正数分别返回 −1, 0 或 1.

例 7
```
Sign[-27.5]
-1
Sign[0]
0
Sign[6.254]
1
```

正整数 n 的阶乘,在数学教科书中记为 $n!$,表示 n 个整数 $1, 2, 3, \cdots, n$ 的连乘积.我们规定 $0! = 1$.对于非整数值的 n, $n!$ 定义为 $\Gamma(n+1)$,这里 Γ 为欧拉伽马函数.

■ **Factorial[n]** 或 **n!** 给出 n 的阶乘.

例 8
```
5!
120
0!
1
Factorial[3.5]
11.6317
```

在 Mathematica 中有一个内置的随机数生成器,这个内置函数在概率论与统计分析(如随机行走与蒙特卡罗方法)中非常有用.

■ **Random[ ]**给出在区间 [0, 1] 上均匀分布的实的伪随机数.

■ **Random[类型]**给出指定类型的均匀分布的伪随机数,其中类型可取值 Integer, Real 或者 Complex.如果取值为 Integer 或 Real,那么返回值介于 0 到 1 之间.如果类型为 Complex,返回值位于由 0 和 1 + i 确定的矩形内.

■ **Random[类型, 范围]**给出位于由范围确定的区间或矩形内均匀分布的伪随机数,其中范围可以是单个数,也可以是两个数,如{a, b} 或 {a + bi, c + di}.单个数 m 等价于{0, m}.

■ **Random[类型, 范围, n]**给出位于由范围确定的区间或矩形内均匀分布的有 n 位有效数字的伪随机数.

例 9 (你所得的答案可能与下面给出的不同)
```
Random[Integer]              ←等概率地返回 0 或 1.
0
Random[Real]                 ←返回介于 0 与 1 之间的有 6 位有效数字的实数.
0.386297
Random[Complex]              ←返回的复数位于对角点为 0 和 1 + i 的矩形内.
0.420851 + 0.382187 i
Random[Real, 5]              ←返回均匀分布在区间 [0,5] 内的实数.
1.83872
Random[Real, {3, 5}]         ←返回均匀分布在区间 [3,5] 内的实数.
3.95386
Random[Real, {3, 5}, 10]     ←返回均匀分布在区间 [3,5] 内的有 10 位有效数字的实数.
```

```
4.014673297
Random[Integer,{1,10}]          ←返回的整数介于1与10之间,概率为1/10.
7
Random[Complex,{2 + I, 5 + 6I}]  ←返回的复数位于对角点为复数2+i和5+6i的矩形内.
2.61319 + 4.30869 i
```

一个正整数为素数是指它只能被1和它本身整除.由于技术上的原因,不认为1是素数,最小的素数是2.

■ **Prime[n]** 给出第n个素数.

例10 求出第7个素数.

```
Prime[7]
17
```

Fibonacci数的定义为

$$f_1 = 1,$$
$$f_2 = 1,$$
$$f_n = f_{n-2} + f_{n-1},\ n \geqslant 3.$$

因此前几个Fibonacci数为1, 1, 2, 3, 5, 8, 13, 21, ….

■ 第n个Fibonacci数是由**Fibonacci[n]**给出的.

例11

```
Fibonacci[7]
13
```

在Mathematica中有三个函数可以把实数转化为附近的整数.

■ **Round[x]** 返回最靠近x的整数.如果x恰好位于两个整数的中间,如5.5,那么Round就返回最靠近的偶数.

■ **Floor[x]** 返回不超过x的最大整数.这个函数有时被称为"最大整数函数",在许多教科书用⌊x⌋表示.

■ **Ceiling[x]** 返回不小于x的最小整数.许多教科书中用⌈x⌉表示它.

例12

```
Round[5.75]
6
Floor[5.75]
5
Ceiling[5.75]
6
```

一个十进制小数可以分为两部分,即整数部分(在小数点左边的部分)和小数部分(在小数点右边的部分).

■ **IntegerPart[x]** 给出x的整数部分(不含小数点).

■ **FractionalPart[x]** 给出x的小数部分(包含小数点).

注意**IntegerPart[x] + FractionalPart[x] = x.**

例 13
```
IntegerPart[4.67]
4
FractionalPart[4.67]
0.67
IntegerPart[4.67] + FractionalPart[4.67]
4.67
```

如果 m 和 n 都是整数,那么存在惟一的整数 q 与 r,使得

$$m = qn + r \qquad 0 \leqslant r < n$$

这个结果就是非常有名的整数相除算法(Division Alogrithm). q 称为商(quotient),而 r 称为余数(remainder). Mathematica 函数 **Quotient** 和 **Mod** 分别返回商与余数.

- **Quotient[m,n]**返回 m 除以 n 得到的商.
- **Mod[m,n]**返回 m 除以 n 得到的余数.

例 14
```
Quotient[17,3]
5
Mod[17,3]
2
```

假设 a 与 b 为两个整数,如果存在整数 k 使得 $a = kb$,则称 b 可整除 a,也称 a 为 b 的倍数.

令 m 与 n 为两个整数,如果 b 同时整除 m 与 n,我们称 b 为 m, n 的公约数. m 与 n 的最大的那个公约数称为 m 与 n 的最大公约数(the greatest common divisor).

如果 a 同时为 m 与 n 的倍数,那么称 a 为 m 与 n 的公倍数. m 与 n 的最小的那个公倍数称为 m 与 n 的最小公倍数(the least common multiple).

- **GCD [m,n]** 返回 m 与 n 的最大公约数.
- **LCM [m,n]** 返回 m 与 n 的最小公倍数.

例 15 求出 48 与 72 的最大公约数与最小公倍数.
```
GCD[48,72]
24
LCM[48,72]
144
```

GCD 与 **LCM** 函数可以推广到多个参数值的情形.

例 16 求出 24, 40 和 48 的最大公约数和最小公倍数.
```
GCD[24, 40, 48]
8
LCM[24, 40, 48]
240
```

算术基本定理保证了每个正整数都可以惟一地分解为素数的乘积.

- 函数 **FactorInteger[n]** 给出 n 的素数分解,同时列出相应的指数.

例 17 **FactorInteger[2381400]**

{{2, 3}, {3, 5}, {5, 2}, {7, 2}}

2 381 400 的素因子为 2, 3, 5, 7, 它们的指数分别为 3, 5, 2, 2. 也就是说 2 381 400 = $2^3 3^5 5^2 7^2$. 这个运算给出的结果是嵌套列表(list), 列表是一个 Mathematica 对象, 用大括号包围起来, 并将在第三章中详细介绍.

为了估计计算的效率, 因此需要能够确定要执行的一个操作或一系列操作的时间.

- **Timing[表达式]** 计算表达式, 并给出一个列表, 由所花费的时间以及所得的结果构成.

这里的计时只考虑了 Mathematica 内核所花费的 CPU 时间, 并没有包含显示到终端上要花费的时间.

例 18 为了计算第 1 000 000 个素数, 内核需要花费多少时间?

Prime[1000000]//Timing ← 这等价于Timing[Prime[1000000]].

{0.51 Second, 15485863}

当然, 实际所用的时间可能与这里给出的不同, 具体与 CPU 的速度有关.

相应于任意底的对数与指数函数也很容易计算.

- **Log[x]**表示自然对数. 如果需要的底不是 e, 而是 b, 那么相应的形式为**Log[b, x]**.
- 函数 **Exp[x]**为自然指数函数. 其他的等价形式为 **E ^ x** 或者 **E^x**. 这时不能用小写字母 e, 但可以用 BasicInput 模板上的符号**e**. 相应于底 **b** 的指数函数形式为**b ^ x**或者 **b^x**.

例 19 计算 ln 100, 即 100 的自然对数.

Log[100]

Log[100]

Log[100]//N

4.60517

> 注意 Mathematica 总是给出精确的答案. 只有当要求时, 才会给出近似值.

例 20 计算 $\log_2 100$.

Log[2, 100]

$\frac{\text{Log}[100]}{\text{Log}[2]}$ ←这是用自然对数表示出来的 $\log_2 x$ 精确值.

Log[2, 100]//N

6.64386

例 21 为了计算 e^2 的近似值, 输入

Exp[2]//N 或者 **E^2//N** 或者**e^2 //N**

7.38906

- 六个基本的三角函数(正弦、余弦、正切、正割、余割和余切)在 Mathematica 中分别用 **Sin, Cos, Tan, Sec, Csc** 和 **Cot** 表示.

Mathematica 假定三角函数的参数值以弧度为单位. 因此如果调用三角函数时参数值单位为角度, 那就必须先把它转化为弧度. 为此需要利用内置常数 **Degree**, 它的值为 $\pi/180$. 在 **BasicInput** 模板上的 ° 符号也可以作此用途.

例 22　60° 等价于 π/3 弧度. 为了计算它的正弦, 需要利用弧度单位, 因此输入

```
sin[π/3]或者 Sin[Pi/3]
```

$\frac{\sqrt{3}}{2}$

如果想用角度计算它的正弦, 那么可以输入

```
Sin[60 Degree] 或者 Sin[60°]
```

$\frac{\sqrt{3}}{2}$

在计算三角函数的幂次时, 必要仔细一些. 在数学书中通常把 sin x 的平方表示为 $\sin^2 x$, 而在 Mathematica 中只能用**Sin[x]²** 或**Sin[x] ^ 2**.

例 23　计算 sin 60° 的平方.

```
Sin[60°]²或者 Sin[60 Degree] ^ 2
```

$\frac{3}{4}$

反三角函数分别是 **ArcSin, ArcCos, ArcTan, ArcSec, ArcCsc** 和 **ArcCot**. 然而这些函数只返回用弧度表示的主值.

例 24

```
ArcSin[1]
```

$\frac{\pi}{2}$

```
ArcCos[Cos[3π]]
```

π

> cos[3π] = -1, 但是 ArcCos[-1]的主值为 π.

双曲函数是指数函数的组合, 它有许多有趣的数学性质. 双曲函数共有六个, 其中最基本的三个函数是

$$\sinh x = \frac{e^x - e^{-x}}{2} \qquad \cosh x = \frac{e^x + e^{-x}}{2} \qquad \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

而另外三个双曲函数是 sech x, csch x 和 coth x, 它们分别是 cosh x, sinh x 和 tanh x 的倒数.

■　六个双曲函数的 Mathematica 表示是 **Sinh, Cosh, Tanh, Sech, Csch, Coth**.

例 25　计算 sinh 2 的近似值.

```
Sinh[2]//N
3.62686
```

反双曲函数分别用 **ArcSinh, ArcCosh, ArcTanh, ArcSech, ArcCsch** 和 **ArcCoth** 表示.

由于 **Cosh** 和 **Sech** 并不是一一对应的, 因此 **ArcCosh** 和 **ArcSech** 对于实参数值, 只返回正值.

例 26

```
ArcSinh[-2]//N
-1.44364
ArcCosh[2]//N
1.31696
```

此时有一条特殊的命令值得注意：

■ **Print**［**表达式**］显示表达式，后接分行符.

■ **Print**［**表达式 1，表达式 2，**…］显示表达式 1，表达式 2，…，然后只接一个分行符.

初一看可能会觉得 **Print** 命令是多余的，因为只要输入任何变量的名称，就会得出它的值.然而，在有些地场合中它是非常有用的(例如，见本章稍后第 2.7 节循环).

例 27 `Print["This prints a line of text."]`

```
This prints a line of text.
```

例 28

```
a = 1 ;b = 2 ;c = 3 ;d = 4; e = 5 ;
Print[a ,b ,c ,d ,e]
12345
```

在 Mathematica 中有一类函数以字母 Q 结尾：

ArgumentCountQ	**AtomQ**	**DigitQ**
EllipticNomeQ	**EvenQ**	**ExactNumberQ**
FreeQ	**HypergeometricPFQ**	**InexactNumberQ**
IntegerQ	**IntervalMemberQ**	**InverseEllipticNomeQ**
LegendreQ	**LetterQ**	**LinkConnectedQ**
LinkReadyQ	**ListQ**	**LowerCaseQ**
MachineNumberQ	**MatchLocalNameQ**	**MatchQ**
MatrixQ	**MemberQ**	**NameQ**
NumberQ	**NumericQ**	**OddQ**
OptionQ	**OrderedQ**	**PartitionsQ**
PolynomialQ	**PrimeQ**	**SameQ**
StringMatchQ	**StringQ**	**SyntaxQ**
TrueQ	**UnsameQ**	**UpperCaseQ**
ValueQ	**VectorQ**	

这些函数用来检测特定的条件，并返回 True(真)或 False(假)值.可以从 Help 菜单中或者像下面例子中那样用 ? 命令找到这些命令的语法.

例 29 **? PrimeQ**

PrimeQ[expr] yields True if expr is a prime number, and yields False otherwise.

(PrimeQ[expr]：当 expr 是素数时，返回真，否则返回假.)

```
PrimeQ[5]
True
PrimeQ[6]
False
```

例 30 **? PolynomialQ**

PolynomialQ[expr, var] yields True if expr is a polynomial in var, and yields False otherwise. PolynomialQ[expr, {var1, …}] tests whether expr is a polynomial in the vari.

(PolynomialQ[expr, var]:如果 expr 为 var 的多项式,返回真,否则返回假. PolynomialQ[expr, {var1, …}]用来判断 expr 是否是 vari 的多项式.)

```
PolynomialQ[x^2 y + x + √Y, x]
True
PolynomialQ[x^2 y + x + √Y, y]
False
```

习题解答

2.5 计算 10 的平方根与立方根的近似值.

解 ☞

√10//N 或者 Sqrt[10] //N

3.16228

∛10//N 或者 10 ^ (1/3) //N

2.15443

2.6 二项式系数定义为$C(n,k)=\frac{n!}{k!\ (n-k)!}$.利用这个定义计算 $C(10,4)$.

解 ☞

$\frac{10!}{4!\ (10-4)!}$或者 Factorial[10]/(Factorial[4] * Factorial[10 - 4])

210

2.7 一个骰子有六个面,编号分别为 1 到 6,每个面出现的概率是相同的.对四次掷骰子的过程进行模似.

解 ☞

```
Random[Integer, {1, 6}]
Random[Integer, {1, 6}]
Random[Integer, {1, 6}]
Random[Integer, {1, 6}]
6
1
5
3
```

2.8 给出介于 π 与 2π 之间的有 15 位有效数字的伪随机数.

解 ☞

```
Random[Real, {π, 2π}, 15]
4.13129131207734
```

2.9 第 27 个 Fibonacci 数是多少?

解 ☞

```
Fibonacci[27]
```

```
196418
```

2.10　证明在 157 与 163 之间没有素数.

解 ☞

```
Prime[37]
157
Prime[38]
163
```

← 通过试验确定 37 这个值.

由于 157 与 163 为相邻的素数,因此在它们之间再没有素数.

2.11　最靠近 $\sqrt{159}$的整数是多少?

解 ☞

```
Round[Sqrt[159]] 或者 √159 //Round
13
```

2.12　$(\pi^2+1)^5$位于哪两个相邻整数之间?

解 ☞

```
Floor[(π² + 1)⁵]
151729
Ceiling[(π² + 1)⁵]
151730
```

因此数 $(\pi^2+1)^5$位于 151 729 与 151 730 之间.

2.13　先用 $x=17$, 再用 $x=\pi$, 计算$\lceil x\rceil-\lfloor x\rfloor$的值.

解 ☞

```
x = 17 ;
Ceiling[x] - Floor[x]
0
x = Pi
Ceiling[x] - Floor[x]
1
```

当 x 为整数时, $\lceil x\rceil-\lfloor x\rfloor$总是等于 0, 而 x 为非整数时, 总是等于 1.

2.14　5 355 与 40 425 的最大公约数与最小公倍数是多少?

解 ☞

```
GCD[5355, 40425]
105
LCM[5355, 40425]
2061675
```

2.15　证明 15, 16 和 30 是互素的.(如果几个数没有不同于 1 的公约数,则称它们是互素的.)

解 ☞

```
GCD[15, 16, 30]
```

1

由于它们的 GCD 等于 1,即它们只有惟一的公约数 1,因此它们是互素的.

2.16 在数论中有一个定理说两个数的 GCD 与 LCM 的乘积等于这两个数的乘积.利用 74 613和 85 085 验证这个定理.

解 ☞

```
a = 74613;
b = 85085;
GCD[a, b] * LCM[a, b]
6348447105
a * b
6348447105
```

显然两个乘积是相同的.

2.17 证明 156 875 438 767 不是素数.

解 ☞

```
FactorInteger[156875438767]
{{53,1}, {2959913939,1}}
```

这就证明了 156 875 438 767 等于素数 53 与 2 959 913 939 的乘积,因此它不是素数.

2.18 在前一个问题中,为了分解 156 875 438 767, Mathematica 花费了多长时间?

解 ☞

`Timing[FactorInteger[156875438767]]` 或者

```
FactorInteger[156875438767]//Timing
{0.06 Second, {{53,1}, {2959913939,1}}}
```

即大约花费了 0.06 秒的时间(具体时间值与计算机有关).

2.19 计算 e^5 的自然对数.

解 ☞

`Log[E^5]` 或者 `Log[E ^ 5]` 或者 `Log[Exp[5]]`

5

2.20 计算 e^5的常用对数(以 10 为底).它的近似值为多少?

解 ☞

`Log[10, E^5]` 或者`Log[10, E ^ 5]` 或者 `Log[10, Exp[5]]`

$$\frac{5}{\text{Log}[10]}$$

```
%//N
2.17147
```

2.21 计算$\sqrt{3}+\sqrt{2}$和$\sqrt{3}-\sqrt{2}$到 50 位有效数字.然后计算它们的乘积.

解 ☞

`a = N[√3 + √2, 50]`

```
3.1462643699419723423291350657155704455124771291873
b = N[√3 - √2, 50]
0.31783724519578224472575761729617428837313337843343
a * b
1.0000000000000000000000000000000000000000000000000
```

2.22 sin 15° 的准确值是多少？给出它的有 20 位小数的近似值.

解 ☞

Sin[15 Degree] 或者 Sin[15°]

$$\frac{-1+\sqrt{3}}{2\sqrt{2}}$$

```
N[%, 20]
0.25881904510252076235
```

2.23 选择介于 0 与 1 之间的一个随机数 x，计算 $\sin^2 x+\cos^2 x$.

解 ☞

```
x = Random[ ];
Sin[x]^2 + Cos[x]^2
1
```

> 在三角函数中，对于所有的 x，$\sin^2 x+\cos^2 x=1$.

2.24 求出介于 $-\pi/2$ 到 $\pi/2$ 之间的一个数，它的正弦为 1/2.

解 ☞

ArcSin[1/2]

$$\frac{\pi}{6}$$

2.25 选择一个介于 0 到 1 之间的随机数，计算 $\cosh^2 x-\sinh^2 x$.

解 ☞

```
x = Random[ ];
Cosh[x]^2 - Sinh[x]^2
1
```

> 双曲函数的性质类似于三角函数：对所有的 x，$\cosh^2 x-\sinh^2 x=1$.

2.26 给出 $\tanh(\ln x)$的等价表示.

解 ☞

Tanh[Log[x]]

$$\frac{-1+x^2}{1+x^2}$$

2.27 一弧度大约等于多少度？

解 ☞

```
N[1/Degree]
57.2958
```

2.28　∞ + 100 000 是多少?

解 ☞

```
∞ + 100000
∞
```

2.29　复数 3 + 4i 的平方根是多少?

解 ☞

$\sqrt{3+4I}$ 或者 `Sqrt[3 + 4I]`

```
2 + i
```

2.30　从 n 个物体中一次取 k 个的排列总数为 $P(n,k)=\frac{n!}{k!}$. 那么从 20 个物体中一次取 10 个的排列总数是多少?

解 ☞

```
n = 20;
k = 10;
```

`n!/k!` 或者 `Factorial[n]/Factorial[k]`

```
67044257280
```

2.31　100 000 的自然对数介于哪两个相邻整数之间?

解 ☞

```
Floor[Log[100000]]
11
Ceiling[Log[100000]]
12
```

所以 ln 100 000 位于 11 和 12 之间.

2.32　62 173 467 除以 9 542 的商和余数分别是多少?

解 ☞

```
Quotient[62173467, 9542]
6515
Mod[62173467, 9542]
7337
```

2.33　求出 1 001 与 1 331 的最大公约数与最小公倍数.

解 ☞

```
GCD[1001, 1331]
11
LCM[1001, 1331]
121121
```

2.34　对 10! 进行素因数分解,计算机需要花费多长时间?

解 ☞

```
FactorInteger[10!]//Timing
{0.01 Second, {{2, 8}, {3, 4}, {5, 2}, {7, 1}}}
```

因此素因数分解结果为 $2^8 3^4 5^7 7^1$，所花费的时间与所用计算机的 CPU 速度有关.

2.35 给出 $\cos\left(\sin^{-1}\left(\frac{x^2}{x^2+1}\right)\right)$ 的代数表示.

解 ☞

```
Cos[ArcSin[x^2/(x^2 + 1)]]
```

$$\sqrt{1-\frac{x^4}{(1+x^2)^2}}$$

2.36 15 485 863 是素数吗?

解 ☞

```
PrimeQ[15485863]
True
```

2.3 基本的算术操作

我们已知道，类似于加法等的基本算术操作是在两个操作数之间插入一个操作符号实现的. 因此 3 与 5 的和是通过输入 3 + 5 得到的. 然而，在某些高级应用中，如果能用函数表示这些操作，就非常有用. 正是处于这个原因，Mathematica 包含了下述函数：

- **Plus[a, b, …]** 计算 a, b, …的和. **Plus[a, b]** 等价于 **a + b**.
- **Times[a, b, …]** 计算 a, b, …的积. **Times[a, b]** 等价于 **a * b**.
- **Subtract[a, b]** 计算 a 与 b 的差. 只能使用两个参数值. **subtract[a, b]** 等价于 **a - b**.
- **Divide[a, b]** 计算 a 与 b 的商. 只能使用两个参数值. **Divide[a, b]** 等价于 **a/b**.
- **Minus[a]** 给出 a 的加法逆元(相反数). **Minus[a]** 等价于 **-a**.
- **Power[a, b]** 计算 a^b. **Power[a, b, c]** 给出 a^{b^c}，等等.

例 31

```
Plus [2, 3, 4]
9
Times[2, 3, 4]
24
Power[2, 3, 4]
2417851639229258349412352
```

如果要查看 Mathematica 对这些函数内部处理的方法，通常使用 **FullForm** 命令.

- **FullForm [表达式]** 显示表达式的内部形式.

FullForm 命令不但可用于算术操作，而且也适用于所有其他类型的 Mathematica 函数操作.

例 32

```
FullForm[a + b + c]
Plus[a, b, c]
FullForm[a - b]
```

```
Plus[a, Times[-1, b]]
FullForm[(a * b) ^ c]
Power[Times[a, b], c]
```

除了上面讨论的标准算术操作符号外，还有几个在特殊场合非常有用的命令：

- **Increment[x]** 或 **x++** 使 x 的值增 1，但返回 x 原来的值.
- **Decrement[x]** 或 **x--** 使 x 的值减 1，但返回 x 原来的值.
- **PreIncrement[x]** 或 **++x** 使 x 的值增 1，并返回 x 的新值.
- **PreDecrement[x]** 或 **--x** 使 x 的值减 1，并返回 x 的新值.
- **AddTo[x, y]** 或 **x+=y** 使 x 的值增加 y，并返回 x 的新值.
- **SubtractFrom[x, y]** 或 **x-=y** 使 x 的值减少 y，并返回 x 的新值.
- **TimesBy[x, y]** 或 **x*=y** 使 x 的值乘以 y，并返回 x 的新值.
- **DivideBy[x, y]** 或 **x/=y** 使 x 的值除以 y，并返回 x 的新值.

为了使 **x++**，**++x**，**x--**，**--x**，**x+=y**，**x*=y** 或 **x/=y** 发挥作用，**x** 必须具有数值. 下面两个例子演示了各种加法命令. 相应的减法、乘法和除法的命令类似.

例 33

```
x = 3;
x++
3      ←返回 x 原来的值
x
4      ←x 的实际值为 4
x++  等价于语句序列
     x
     x = x + 1;
```

```
x = 3;
++x
4      ←返回 x 的新值
x
4      ←x 的实际值为 4
++x 等价于语句 x = x + 1
```

例 34

```
x = 3; y = 4;
x + y
7      ←返回新的和值
x
4      ←x 的值没有改变
```

```
x = 3; y = 4;
x += y
7      ←返回新的和值
x
4      ←x 的新值为 7
x += y 等价于语句 x = x + y
```

习 题 解 答

2.37 Mathematica 是如何计算表达式 $a + bc/d$ 的？

解 ☞

```
FullForm[a + b * c/d]
Plus[a, Times[b, c, Power[d, -1]]]
```

2.38 在 Mathematica 内部是如何处理 Minus[x]函数的呢？

解 ☞

```
FullForm[Minus[x]]
Times[-1, x]
```

2.4 字符串

字符串也就是(有序的)字符序列.字符串的值并不是数,通常被用来做为表格、图形和其他结果的标签.

在 Mathematica 中,字符串被包围在双引号内.因此 "abcde" 就是有五个字符的字符串.注意不要把 "abcde" 与 abcde 混在一起,因为后者表示一个符号,而并不是字符串.

在 Mathematica 中有许多操作字符串的命令.

- **StringLength[字符串]** 返回在字符串中的字符个数.
- **StringJoin[字符串 1, 字符串 2, …]** 或者 **字符串 1 < >字符串 2 < >…** 把两个或多个字符串组合成一个新字符串,合并后的长度等于原来每个字符串长度的总和.
- **StringReverse[字符串]** 使字符串反序.

StringDrop 从字符串中删除字符,这条命令有五种形式.

- **StringDrop[字符串, n]** 返回从字符串中删除前 n 个字符后的结果.
- **StringDrop[字符串, -n]** 返回从字符串中删除后 n 个字符的结果.
- **StringDrop[字符串, {n}]** 返回从字符串中删除第 n 个字符的结果.
- **StringDrop[字符串, {-n}]** 返回从字符串中删除倒数第 n 个字符的结果.
- **StringDrop[字符串, {m, n}]** 返回从字符串中删除从第 m 个到第 n 个字符后的结果.

StringTake 命令从字符串提到字符.它的形式类似于 StringDrop:

- **StringTake[字符串, n]** 返回字符串中的前 n 个字符.
- **StringTake[字符串, -n]** 返回字符串中的后 n 个字符.
- **StringTake[字符串, {n}]** 返回字符串中的第 n 个字符.
- **StringTake[字符串, {-n}]** 返回字符串中的倒数第 n 个字符.
- **StringTake[字符串, {m, n}]** 返回字符串中的从第 m 个到第 n 个字符.

例 35 在这个例子中我们定义 **strng ="abcdefg"**.在下面的输出都显示在命令的右边.(注意这里为了避免与 Mathematica 符号'String'混淆,并没有使用'string')

```
strng = "abcdefg"                 abcdefg
strng <>"hijklmnop"               abcdefghijklmnop
StringLength[strng]               7
StringReverse[strng]              gfedcba
StringDrop[ strng, 2]             cdefg
StringDrop[ strng, -2]            abcde
StringDrop[ strng, {2}]           acdefg
StringDrop[ strng, {-2}]          abcdeg
StringDrop[ strng, {2, 5}]        afg
StringTake[ strng, 2]             ab
StringTake[ strng, -2]            fg
StringTake[ strng, {2}]           b
StringTake[ strng, {-2}]          f
StringTake[ strng, {2, 5}]        bcde
```

StringInsert 函数可以向已有字符串中插入字符.

- **StringInsert[字符串 1, 字符串 2, n]** 生成一个新的字符串,其构成为字符串 2 插入在字符串 1 的第 n 个字符前.
- **StringInsert[字符串 1, 字符串 2, -n]** 生成一个新的字符串,其构成为字符串 2

插入在字符串 1 的倒数第 n 个字符前.

- **StringInsert [字符串 1, 字符串 2, {n1, n2, …}]** 生成一个新的字符串,其构成为多份字符串 2 插入在字符串 1 的第 n1, n2, …个字符前.

StringReplace 函数可以用另外的字符串替换给定字符串的一部分.

- **StringReplace [字符串, 字符串 1 →新字符串 1]** 在字符串中只要有字符串 1 出现,就有新字符串 1 替换.
- **StringReplace [字符串, {字符串 1 →新字符串 1, 字符串 2 →新字符串 2, …}]** 在字符串中只要有字符串 1 出现,就有新字符串 1 替换,同样只要有字符串 2 出现,就有新字符串 2 替换,….
- **StringPosition [字符串, 子字符串]** 返回一个列表,由子字符串在字符串中出现的所有开始与结束位置组成.(列表将在第三章详细介绍.)

例 36
```
strng1 = "abcdefg";
strng2 ="123";
StringInsert [strng1, strng2, 3 ]
ab123cdefg
StringInsert [strng1, strng2, -3 ]
abcde123fg
StringInsert [strng1, strng2, {1, 3, 5, 7}]
123ab123cd123ef123g
StringReplace[ strng1, "ab" ->"AB"]
ABcdefg
StringReplace[ strng1, {"ab" ->"AB", "fg" ->"FG"}]
ABcdeFG
```

例 37
```
strng ="abcxabcxxabcxxxabc";
StringLength[strng]
18
StringPosition[strng, "abc"]
{{1, 3}, {5, 7}, {10, 12}, {16, 18}}
```

2.5 赋值、替换与逻辑关系

在所有的程序设计语言中,为了把计算结果传递给一个符号,以便稍后再使用,都提供了赋值功能.Mathematica 提供了两种类型的赋值,但是到底在哪种情况下使用哪种赋值语句,经常会使人迷惑.

- **lhs = rhs** 是一个即时赋值语句,这里在进行赋值时,就对 rhs 进行计算.
- **lhs := rhs** 是一个延时赋值语句,以后每次调用时才计算 rhs.

在许多情况中这两种赋值语句产生相同的结果,然而在有些情况中还是需要细心处理的.在下面给出的例子中应用了后面章节将讨论的概念.但因这些例子是不解自明的,故比较容易理解.

例 38 当递归定义函数(第 2.9 节)时,就必须使用 := 运算符.例如,
```
f[0] = 1;
f[n] := n f[n - 1]
```
给出 n 的阶乘.因此直到给出 n 的值时,Mathematica 才计算 f[n]的值,所以必须用延时赋值

符号 := . 如果使用 = 就会导致递归出错.

例 39 当定义分段函数(第 2.9 节)时,我们必须用 := . 例如,

```
g[x_] := x^2/;x≥0          ←/;是一个条件.只有当 x≥0 时才进行赋值.
g[x_] := -x^2/;x<0
```

如果使用 = 就会引起麻烦,因为直到给出 x 的值时,Mathematica 才能确定取的是哪一个分支.

例 40 这样你就有可能以为 := 赋值更具有一般性,可以应用到任意场合.在某种意义上讲,这是正确的,但在有些情况中,我们应该使用 = . 下面给出一个极端但也颇有道理的例子,定义

$$F[x_]:=\int_0^x tE^t Sin[t]\,dt$$

每次需要计算 F 的一个值时,Mathematica 就要进行几次"分部积分"计算.不妨想像一下在需要得到 F 在许多不同点上的函数值时所遇到的问题,例如在指令 `Plot[F[x],{x, 0, 5}]`. 它要利用 F 在 0 到 5 之间的许多值作图.每次计算 F 的值时,就要从头开始每次利用分部积分重新计算积分,因此显示图形就要花费相当长的时间.而使用 = 就会使图形的显示相当迅速.

有时候,想计算表达式的值,而又不想给某个符号赋值,那么可以用替换符号/.

例 41 假设要计算 $x=3$ 时表达式 x^2+5x+6 的值,而又不想给 x 赋值.

```
x^2 + 5x + 6 /.x→3
30
? x
Global`x             ← x 仍然没有定义.
```

/. 也可以用来把一个表达式替换为另一个表达式.如果使用了花括号的话,可以一次进行多个替换.

例 42

```
√(2x+3) + (2X+3)^2/.2x + 3→3y + 5
   √(3y+5) + (3y+5)^2
```

例 43

```
2x+3y/.{y→x, x→y}
3x+2y
```

注意 不要把 = 与逻辑等号 == 混淆. lhs == rhs 的值为 True 当且仅当 lhs 和 rhs 有相同的值,否则它的值为 False. 在方程求解(第六章)中将大量用到逻辑等式.

除此之外,还有一些可以应用的逻辑关系.下面汇总了这些逻辑关系.

- **Equal[x, y]** 或 **x == y** 的值为真当且仅当 x 与 y 有相同的值.
- **Unequal[x, y]** 或 **x != y** 或 **x ≠ y** 的值为真当且仅当 x 与 y 的值不同.
- **Less[x, y]** 或 **x < y** 的值为真当且仅当在数值上 x 小于 y.
- **Greater[x, y]** 或 **x > y** 的值为真当且仅当在数值上 x 大于 y.
- **LessEqual[x, y]** 或 **x <= y** 或 **x ≤ y** 的值为真当且仅当在数值上 x 小于或等于 y.
- **GreaterEqual[x, y]** 或 **x >= y** 或 **x ≥ y** 的值为真当且仅当在数值上 x 大于或等于 y.

注意 Equal 与 Unequal 可以比较数值量,也可以比较非数值量,而 Less, Greater, LessEqual 与 GreaterEqual 则只能进行数值量比较.

例 44

```
1 == 2        1 != 2        1 <=2        a + a == 2a
False         True          True         True
2 == 2        2 != 2        2 <=2        a <a
True          False         True         a<a
```

在 Mathematica 中也包含下述逻辑操作.

- **And**[p, q] 或 p&&q 或 p ∧q, 当 p 与 q 的值都为 True 时, 它的值为 **True**, 否则为 False.
- **Or**[p, q] 或 p||q 或 p ∨q, 当 p 或 q 的值(或两者同时)为 True 时, 它的值为 True, 否则为 False.
- **Xor**[p, q], 当 p 或 q 的值(但不同时)为 True 时, 它的值为 True, 否则为 False.
- **Not**[p] 或 ! p 或 ¬p, 当 p 的值为 False 时, 它的值为 True; 当 p 的值为 True 时, 它的值则为 False.
- **Implies**[p, q] 或 p ⇒q, 如果 p 的值为 True, q 的值为 False, 则它的值为 False, 否则它的值为 True.

注意　可以用序列按键 [ESC] + [=] + [>] + [ESC] 得到 ⇒.

逻辑表达式可以用 **LogicalExpand** 命令进行比较, 这条命令把逻辑表达式转化为析取范式.

例 45　利用 Mathematica 验证逻辑分配律: p∧(q∨r) = (p∧q)∨(p∧r).

```
lhs = p&&(q||r);
rhs = (p&&q)||(p&&r);
lhs == rhs
(p && (q || r)) == (p && q || p && r)
LogicalExpand[lhs] ==LogicalExpand[rhs]
True
```

习 题 解 答

2.39　我们将在第七章中介绍 Mathematica **命令 Expand[表达式]**, 这条命令对表达式进行代数展开. 分别用 = 和 := 把两个符号 a 与 b 都定义成 Expand[(x + 1) ^ 3]. 然后令 x = u + v, 再计算 a 与 b, (体会两者之间的差别.)

解 ☞

```
a = Expand[(x + 1) ^ 3]
1 + 3x + 3x^2 + x^3                      ←马上展开表达式
b := Expand[(x + 1) ^ 3]                 ←直到调用 b 时才会进行展开
x = u + v
u + v
a
1 + 3(u + v) + 3(u + v)^2 + (u + v)^3    ← 在展开后把 x 替换为 u + v
b
1 + 3u + 3u^2 + u^3 + 3v + 6uv +
3u^2v + 3v^2 + 3uv^2 + v^3               ←在展开之前把 x 替换为 u + v
```

2.40 Together 命令将在第七章中讨论，它把两个分数组合利用通分组成有公分母的一个分数.分别用 = 和 := 把符号 y 与 z 定义成 Together[a + b]，然后令 a = 1/x，b = 1/(x + 1)，再计算 y 与 z.

解 ☞

```
y = Together[a + b]
a + b                        ←此时 a 与 b 不是分数，因此 Together 不运行.
z := Together[a + b]
a = 1/x;
b = 1/(x + 1);
y
```

$\frac{1}{x}+\frac{1}{1+x}$ ←由于 Together 是在引入分数之前执行，因此 Together[a + b] = a + b.

```
z
```

$\frac{1+2x}{x(1+x)}$ ←Together 是在引入分数之后执行的.

2.41 Mathematica 命令 **Simplify[表达式]** 会尝试简化代数表达式.输入 **a = Simplify[poly]** 和 **b := Simplify[poly]**，然后令 **poly = x² + 2x + 1**. 计算 **a** 与 **b** 的值，并解释结果的差别.

解 ☞

```
a = Simplify[poly];
b := Simplify[poly];
poly = x^2 + 2x + 1;
a
```

$1 + 2x + x^2$

```
b
```

$(1 + x)^2$

由于 a 是马上进行计算的，所以它的值就是符号 poly 的简化形式，仍为 poly. 然后 poly 被替换为 $x^2 + 2x + 1$. 另一方面，直到最后一行调用时 b 才进行简化，因此 Mathematica 就对给定多项式进行简化.

2.42 在表达式 $x^2 + 5x + 6$ 中用 $x^2 + 2x + 3$ 替换 x.

解 ☞

```
x^2 + 5x + 6 /. x -> x^2 + 2x + 3
```

$6 + 5(3 + 2x + x^2) + (3 + 2x + x^2)^2$

2.43 在表达式 $(x + y + z)^2$ 中用 $x + 1$ 替换 y，用 $x + 2$ 替换 z.

解 ☞

```
(x + y + z)^2 /. {y -> x + 1, z -> x + 2}
```

$(3 + 3x)^2$

2.44 利用 Mathematica 验证德·摩尔根(De Morgan)定律：

$$\neg(p\wedge q) = \neg p\vee\neg q \text{ 与 } \neg(p\vee q) = \neg p\wedge\neg$$

解 ☞ LogicalExpand[! (p&&q)] ==LogicalExpand[!p ||!q]
True
LogicalExpand[!(p || q)] ==LogicalExpand[!p&&!q]
True

2.6 和与积

和与积在数学中是非常重要的，Mathematica 使得和与积的计算相当简单．与许多计算机语言不同，这时的初始化是自动进行的，而且语法很容易使用，尤其是如果使用 **BasicInput** 模板的话，更是如此．任何符号都可以做为求和的指标（在下面的描述中使用的是 i）．如果使用了 increment 的话，指标可以负增长．

- **Sum**[a[i], {i, imax}] 计算和 $\sum_{i=1}^{imax} a[i]$.
- **Sum**[a[i], {i, imin, imax}] 计算和 $\sum_{i=imin}^{imax} a[i]$.
- **Sum**[a[i], {i, imin, imax, increment}] 计算和 $\sum_{i=imin}^{imax} a[i]$，步长为 increment.

例 46 为了计算前 20 相邻整数的平方和，输入

Sum[i^2, {i, 1, 20}] 或者 $\sum_{i=1}^{20} i^2$

2870

> 注意，即使 Mathematica 接受形式为 Sum[i^2, {i,20}] 的输入，但出于明了的考虑，建议使用初始值 1.

例 47 计算和 $\frac{1}{15}+\frac{1}{17}+\frac{1}{19}+\cdots+\frac{1}{51}$.

Sum[1/i, {i, 15, 51, 2}]

$\frac{6350139147580604419 3}{9684514075768739707 5}$

- 数值近似值可以用 **NSum** 得到．它的语法与 **Sum** 一样．

例 48 给出和 $\frac{1}{15}+\frac{1}{17}+\frac{1}{19}+\cdots+\frac{1}{51}$ 的近似值.

NSum[1/i, {i, 15, 51, 2}]

0.6557

求和限可以是无穷．Mathematica 采用复杂的方法计算无穷和．

例 49 计算 $\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\cdots$.

Sum[1/i^2, {i, 1, Infinity}] 或者 $\sum_{i=1}^{\infty}\frac{1}{i^2}$

$\frac{\pi^2}{6}$

二重和可以利用下述语法得到，也可以更方便地两次点击 **BasicInput** 模板上的 $\sum$ 符号

实现.这也同样可以扩展到三重和、四重和甚至更多重和.

- **Sum[a[i, j], {i, imax}, {j, jmax}]** 计算和 $\sum_{i=1}^{imax}\sum_{j=1}^{jmax} a[i,j]$.
- **Sum[a[i, j], {i, imin, imax}, {j, jmin, jmax}]** 计算和. $\sum_{i=imin}^{imax}\sum_{j=jmin}^{jmax} a[i,j]$.
- **Sum[a[i, j], {i, imin, imax, i_increment}, {j, jmin, jmax, j_increment}]** 计算和 $\sum_{i=imin}^{imax}\sum_{j=jmin}^{jmax} a[i,j]$,步长分别为 i_increment 与 j_increment.
- **NSum** 与 **Sum** 具有同样的语法,工作方式类似,但给出的是数值近似值.

例 50 计算$\left(\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}\right)+\left(\frac{2}{1}+\frac{2}{2}+\frac{2}{3}+\frac{2}{4}\right)+\left(\frac{3}{1}+\frac{3}{2}+\frac{3}{3}+\frac{3}{4}\right)$.

Sum[i/j, {i, 1, 3}, {j, 1, 4}] 或者 $\sum_{i=1}^{3}\sum_{j=1}^{4}\frac{i}{j}$

$\frac{25}{2}$

正与 Sum 计算和一样,Mathematica 函数 Product 计算乘积.它的语法与 **Sum** 的相同.

- **Product[a[i], {i, imax}]** 计算乘积 $\prod_{i=1}^{imax} a[i]$.
- **Product[a[i], {i, imin, imax}]** 计算乘积 $\prod_{i=imin}^{imax} a[i]$.
- **Product[a[i], {i, imin, imax, increment}]** 计算乘积 $\prod_{i=imin}^{imax} a[i]$,步长为 increment.
- **NProduct** 的语法与上面一样,返回在 Product 中给出的乘积的近似值.

多重乘积也很容易计算.下面列出了二重乘积的语法,它可以推广到三重、四重乘积,等等.

- **Product[a[i, j], {i, imax}, {j, jmax}]** 计算乘积 $\prod_{i=1}^{imax}\prod_{j=1}^{jmax} a[i,j]$.
- **Product[a[i, j], {i, imin, imax}, {j, jmin, jmax}]** 计算和 $\prod_{i=imin}^{imax}\prod_{j=jmin}^{jmax} a[i,j]$.
- **Product[a[i, j], {i, imin, imax, i_increment}, {j, jmin, jmax, j_increment}]**

计算和 $\prod_{i=imin}^{imax}\prod_{j=jmin}^{jmax} a[i,j]$,步长分别为 i_increment 与 j_increment.

例 51 计算从 4 到 9 的相邻整数的乘积.

Product[i, {i, 4, 9}]或者 $\prod_{i=4}^{9} i$

60480

例 52 对于二项式系数 $C(n,k)=\frac{n!}{k!\,(n-k)!}$,为了更高效地计算它的值,可以把它表示为 $\left(\frac{n}{k}\right)\left(\frac{n-1}{k-1}\right)\left(\frac{n-2}{k-2}\right)\cdots\left(\frac{n-k+1}{1}\right)$.利用这个表示计算 $C(10, 4)$.

```
n = 10;
k = 4;
Product[(n-i)/(k-i), {i, 0, k-1}]
```
或者 $\prod_{i=0}^{k-1}\frac{n-i}{k-i}$

210

习题解答

2.45 计算前 25 个素数的和.

解 ☞

Sum[Prime[k], {k, 1, 25}] 或者 $\sum_{k=1}^{25}$ **Prime[k]**

1060

2.46 计算从 15 到 30(含 15 与 30)的整数平方和的平方根.

解 ☞

Sqrt[Sum[k^2, {k, 15, 30}]] 或者 $\sqrt{\sum_{k=15}^{30} k^2}$

$2\sqrt{2110}$

2.47 计算无穷和 $1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\cdots$.

解 ☞

Sum[1/2 ^ i, {i, 0, Infinity}] 或者 $\sum_{i=0}^{\infty} \frac{1}{2^i}$

2

2.48 计算和 $\frac{1}{2}+\frac{2}{3}+\frac{3}{4}+\cdots+\frac{99}{100}$.

解 ☞

$\sum_{i=1}^{99} \frac{i}{i+1}$

$$\frac{264414864639329557497913717698145082779489}{2788815009188499086581352357412492142272}$$

2.49 给出从 1 到 n 相邻整数平方和的一般公式.

解 ☞

Sum[k ^ 2, {k, 1, n}] 或者 $\sum_{k=1}^{n} k^2$

$\frac{1}{6} n(1+n)(1+2n)$ ← Mathematica 记住了这些标准公式.

2.50 计算前 100 个素数的和.

解 ☞

$\sum_{i=1}^{100}$ **Prime [i]** 或者 **Sum[Prime[i], {i, 1, 100}]**

24133

2.51 计算从 2 到 20 的整数的自然对数乘积.给出有 20 位有效数字的近似值.

解 ☞

```
N[Product[Log[i], {i, 2, 20}], 20]
```

$1.3632878207490815857\times10^{6}$

2.52 计算和 $1+\left(1+\frac{1}{2}\right)+\left(1+\frac{1}{2}+\frac{1}{3}\right)+\cdots+\left(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{20}\right)$.

解 ☞

`Sum[1/j, {i, 1, 20}, {j, 1, i}]` 或者 $\sum_{i=1}^{20}\sum_{j=1}^{i}\frac{1}{j}$

$\frac{41054655}{739024}$

2.53 计算$\left(1+\frac{1}{2}\right)\left(1+\frac{1}{2}+\frac{1}{3}\right)\cdots\left(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{10}\right)$的近似值.

解 ☞

`NProduct[Sum[1/j, {j, 1, i}, {i, 2, 10}]]` 或者 $\prod_{i=2}^{10}\sum_{j=1}^{i}\frac{1}{j}$`//N`

1871.44

2.7 循 环

有时候我们可能需要重复一个操作或一组操作几次.虽然 Mathematica 提供了利用 Sum 与 Product 命令方便地计算和与积的功能,但有时候处理的工作还是需要利用循环的方法.在 Mathematica 中提供了三种基本的循环函数:**Do**, **While** 与 **For**.

- **Do[表达式, {k}]** 严格地计算表达式 **k** 次.
- **Do[表达式, {i, imax}]** 计算表达式 imax 次,其中 i 的值从 1 变到 imax,每次步长为 1.
- **Do[表达式, {i, imin, imax}]** 当 i 的值从 imin 变到 imax,步长为 1 时,每次都计算表达式.
- **Do[表达式, {i, imin, imax, increment}]** 当 i 的值从 imin 变到 imax,步长为 increment 时,每次都计算表达式.
- **Do[表达式, {i, imin, imax}, {j, jmin, jmax}, …]** 当 i 的值从 imin 变到imax,步长为 1,j 的值从 jmin 变到 jmax,步长为 1 时,每次都计算表达式.当 j 完成一个循环后,i 的值增加 1,依此类推.这就是所谓的嵌套 Do 循环.
- **Do[表达式, {i, imin, imax, i_increment}, {j, jmin, jmax, j_increment}, …]** 形成一个嵌套 Do 循环,这时步长为指定值.

例 53

```
Do[Print["This line will be repeated 5 times."], {5}]
This line will be repeated 5 times.
This line will be repeated 5 times.
This line will be repeated 5 times.
This line will be repeated 5 times.
This line will be repeated 5 times.
```

例 54 下面这个例子计算从 5 到 25 相邻奇数的和.(当然用 Sum 命令要更方便一些.)

```
mysum = 0;                          ←mysum 的初始化.这一步是非常重要的.
Do[mysum = mysum + k, {k, 5, 25, 2}]
```

```
mysum
165
```

例 55　下面这个例子计算所有分子与分母为不超过 5 的正整数的分数的和.

```
fracsum = 0;
Do[fracsum = fracsum + i/j,{i, 1, 5},{j, 1, 5}]
fracsum
137
 4
```

■ **While[条件, 表达式]** 先计算条件,然后再计算表达式,重复这个过程直到条件的值为 False.

如果表达式是由多个语句组成的,那么它们之间用分号隔开.

例 56

```
n = 1; While[n < 6, Print[n]; n = n + 1]
1
2
3
4
5
```

可以用 n++ 代替 n = n + 1

■ **For[初始化, 检测条件, 增量, 表达式]** 执行初始化语句,然后重复计算表达式与增量,直到检测条件的值为 False.

这里的计算顺序是检测条件,表达式,然后是增量. 只要检测条件的值为 False, For 循环就会结束. 如果初始化、检测条件,增量和表达式中有的由多个语句组成的,那么用分号分开相邻语句.

例 57

```
For[i = 1, i ≤ 5, i = i + 1, Print[i]]
1
2
3
4
5
```

另外,虽然 **If** 指令不是循环语句,但它经常与其他的循环命令合用.

■ **If[条件, 真, 假]** 计算条件,并且当它的条件值为 True 时执行真语句,如果条件值为 False,则执行假语句.

■ **If[条件, 真]** 计算条件,并且当它的条件值为 True 时执行真语句,如果条件值为 False,则不执行任何语句,返回值为 Null.

■ **If[条件, , 假]** 计算条件,并且当它的条件值为 False 时执行假语句,如果条件值为 True,则不执行任何语句,返回值为 Null.(注意这里有两个相邻的逗号.)

■ **If[条件, 真, 假, 空]** 计算条件,并且当它的条件值为 True 时执行真语句,如果条件值为 False,则执行假语句,如果条件的值既不是 True 也不是 False,则执行空语句.

例 58

```
If[2 == 2, Print["a"], Print["b"]]        ←2 == 2 的值为 True.
a
If[2 == 3, Print["a"], Print["b"]]        ←2 == 3 的值为 False.
```

```
b
If[7, Print["a"], Print["b"], Print["c"]]        ←7既不是True,也不是False.
c
```

下面这个例子中把素数与非素数分开,从而演示了在 Do 循环中如何使用 If 指令的方法.

例 59 `Do[If[PrimeQ[k], Print[k], Print[" ", k]], {k, 1, 20}]`

```
    1
2
3
    4
5
    6
7
    8
    9
    10
11
    12
13
    14
    15
    16
17
    18
19
    20
```

习题解答

2.54 利用 Do 循环计算 10!.

解答

```
fact = 1;
n = 10;
Do[fact = fact *k, {k, n}]
fact
3628800
```

2.55 利用 While 循环计算 10!.

解答

```
fact = 1;
n = 10;
While[n>0, fact = fact *n; n--]
fact
```

```
3628800
```

2.56 利用 For 循环计算 10!.

解 ☞

```
For[fact = 1; n = 1, n ≤10, n + +, fact = n * fact]
fact
3628800
```

2.57 显示出从 1 到 20 的整数中不是 2, 3, 5 倍数的那些整数.

解 ☞

```
Do[If[Mod[k, 2] == 0 || Mod[k, 3] = = 0 || Mod[k, 5] == 0, Print[k]], {k, 1,
20}]
1
7
11
13
17
19
```

2.58 对于从 1 到 10 之间的每个数 k, 如果 k 为偶数, 显示出它的一半, 如果 k 为奇数, 显示它的两倍.

```
Do[If[EvenQ[k], Print[k/2], Print[2 k], {k, 1, 10}]
2
1
6
2
10
3
14
4
18
5
```

2.8 绘图简介

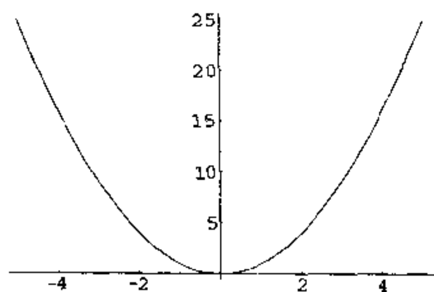
通过观察函数图形, 可以对函数的性态有一目了然的掌握, 而且函数图形对某些问题的求解也有很大帮助. Mathematica 提供了几个功能相当强大的绘图命令, 而且这些命令也非常容易使用. 为了设置输出结果的样式, 这些命令同时提供了一大组选项, 但在本节我们只应用几个由 Mathematica 缺省设置的最基本样式. 关于图形命令的更详细讨论, 请见第四章和第五章.

Plot 命令绘制二维图形.

- **Plot[f[x], {x, xmin, xmax}]** 在区间 xmin ≤ x ≤xmax 上绘制函数 $f(x)$ 的二维图形.
- **Plot[{f[x], g[x]}, {x, xmin, xmax}]** 在同一坐标系中绘制两个函数的图形. 这可以自然地推广到同时绘制三个或更多函数图形的情形.

例 60 在区间 $-5\leqslant x\leqslant 5$ 上绘制 $y=x^2$ 的图形.

```
Plot[x^2, {x, -5, 5}]
```



- Graphics -

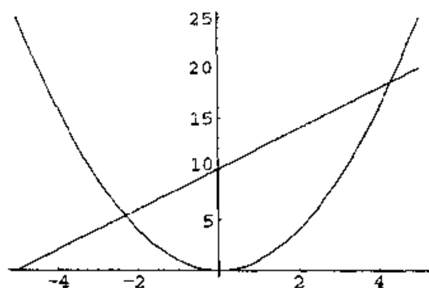
文本"- Graphics -"将会出现在图形的后面(这可能有点儿令人讨厌),因此可以在命令的后面加上分号(;)去掉它.

习 题 解 答

2.59 在同一坐标系内画出 $y=x^2$, $y=2x+10$, $-5\leqslant x\leqslant 5$ 的图形.

解 ☞

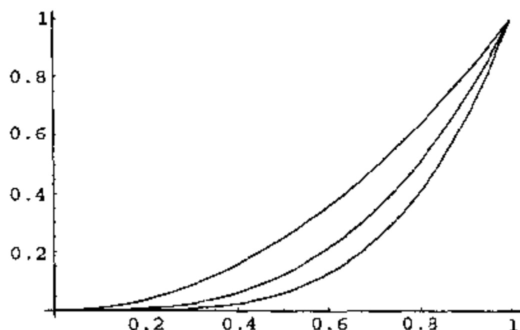
```
Plot[{x^2, 2 x + 10}, {x, -5, 5}];        ← 注意这里使用了分号.
```



2.60 在同一坐标系中画出 $y=x^2$, $y=x^3$, $y=x^4$, $0\leqslant x\leqslant 1$ 的图形.

解 ☞

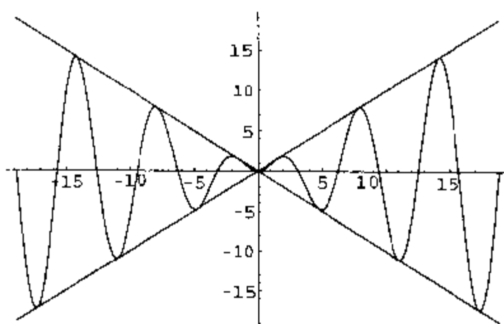
```
Plot[{x^2, x^3, x^4}, {x, 0, 1}];
```



2.61 在同一坐标系中画出区间 $-6\pi\leqslant x\leqslant 6\pi$ 中函数 $y=x$, $y=-x$, $y=x\sin x$ 的图形.

解 ☞

```
Plot[{x, -x, x Sin[x]}, {x, -6π, 6π}];
```



2.9 用户定义的函数

假设希望定义一个单变量函数 f,以 x 为变量,那么可以书写为

f[x_] = ……

或者

f[x_] := ……

其中左边的定义的告诉 Mathematica 当给定 x 的值时,如何计算 f 的值.在定义中可以使用任意合法的 Mathematica 运算,以及引用内置函数.

注意在定义的左边紧接在 x 右面的下划线.这个符号相当重要,它是告诉 Mathematica 这个变量为哑元的惟一办法,所谓哑元就是可以用任何(数值的或符号的)表达式替换的变量.

例 61 **f[x_] = x² + x³;**

f[2]

12

f[2 x]

$4x^2 + 8x^3$

f[Exp[x]]

$e^{2x} + e^{3x}$

f[λ]

$\lambda^2 + \lambda^3$

分段函数可以用 /;条件语句进行定义.即只要输入

f[x_]:=表达式 /; 条件

就会使当且仅当条件值为真时,f[x] 的值为表达式.

注意 在这种应用中,必须使用 := 赋值符号.

例 62 定义函数 $f(x)=\begin{cases}x^2, & \text{如果 } x\leqslant 2,\\ 8-2x, & \text{如果 } x>2.\end{cases}$

f[x_] := x² /;x⩽2

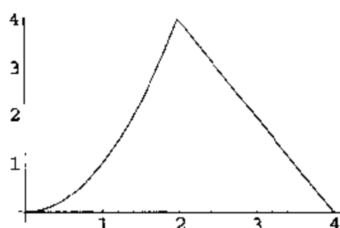
f[x_] := 8 - 2x /; x>2

f[-4]

16

f[4]

```
0
Plot[f[x], {x, 0, 4}];
```



有时候要递归定义函数,即先给出函数的一个或多个值,然后依次用前面的值定义后面的值.

例 63 Fibonacci 数列是通过递归来定义的,即令 $f(1)=1$, $f(2)=1$, 而对于 $n\geqslant 3$, $f(n)=f(n-2)+f(n-1)$. 利用这个定义计算第 25 个 Fibonacci 数是多少.

```
f[1] = 1;
f[2] = 1;
f[n_] := f[n-2]+f[n-1]
f[25]
75025
```

这里有一点儿非常重要,那就是这里必须要用:=,如果改成=看看会出现什么情况.

你可能会发现为了计算这个数,需要花费很长的时间. 为了确认这一点,我们统计一下这个运算花费的时间.(你所得的时间可能与这里给出的不同,具体与计算机速度有关.)

```
f[25]//Timing
{18.18 Second, 75025}
```

这里并没有保存计算的中间结果. 每次计算 f[n]时都需要计算 f[n-1]与 f[n-2],而这每一个都需要计算 f 的一直下溯到 f[3]的每个值. 由于 f 的每个值都是递归利用f[1]与 f[2]的值计算得出的,因此为了计算 f[25],需要利用相当多的迭代. 为了解决这个问题,我们可以在内存中保存计算出来的 f 的每个值,这样在需要时就可以马上找到.

例 64

```
f[1] = 1;
f[2] = 1;
f[n_] := f[n] = f[n-2]+f[n-1]
```

←这使 Mathematica 保存每个 f[n]的值,可以输入? f 来确认这一点.

```
f[25]//Timing
{0. Second, 75025}
```

两个或多个变量的函数也可以用类似方式加以定义,其语法是自明的.

例 65

```
f[x_, y_] = x^2 + y^3;
f[2, 3]
31
f[3,2]
17
```

例 66

```
g[x_, y_, z_] = x + y + z;
```

```
g[2, 3, 4]
14
```

习题解答

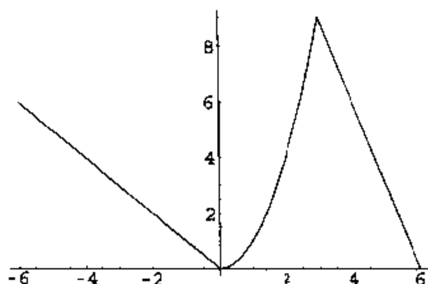
2.62　把 $f(x)$ 定义为多项式 $x^5+3x^4-7x^2+2$，并计算 $f(2)$.

解 ☞

```
f[x_] = x^5 + 3x^4 - 7x^2 + 2
2 - 7x^2 + 3x^4 + x^5
f[2]
54
```

2.63　令 $f(x)=\begin{cases}-x, & \text{如果 } x\leqslant 0,\\ x^2, & \text{如果 } 0<x\leqslant 3,\\ 18-3x, & \text{如果 } x>3,\end{cases}$ 画出 $f(x)$在 $-6\leqslant x\leqslant 6$ 中的图形.

解 ☞

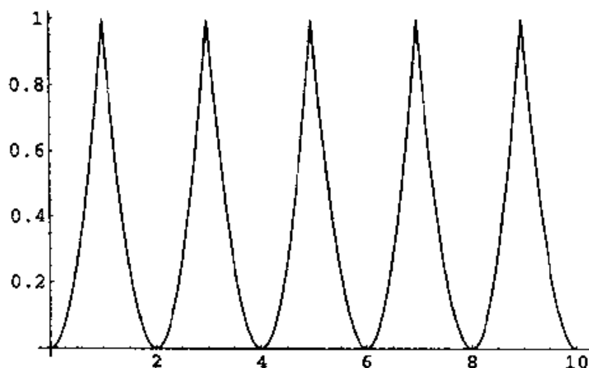


```
f[x_] := -x/; x<=0
f[x_] := x^2/; 0<x<=3
f[x_] := 18 - 3x/; x>3
Plot[f[x], {x, -6, 6}];
```

2.64　设 $f(x)$ 定义在区间 $[a, b]$ 上，f 的周期 $b-a$ 的周期扩展就是如下定义的函数 F：

$$F(x)=\begin{cases}f(x), & \text{如果 } a\leqslant x\leqslant b,\\ f(x-(b-a)), & \text{否则}.\end{cases}$$

设当 $-1\leqslant x\leqslant 1$ 时 $f(x)=x^2$. 绘制 f 的周期为 2 的周期扩展在 $x=0$ 到 $x=10$ 之间的图形.



解 ☞

```
f[x_] = x^2;
extension[x_] := f[x]/; -1 ≤ x ≤ 1;
extension[x_] := extension[x - 2]/;x > 1;
Plot[extension[x], {x, 0, 10}];
```

2.65 定义函数 $f(n)$: $f(1)=1$, $f(2)=2$, $f(3)=3$, 如果 $n \geqslant 4$, $f(n)=f(n-3)+f(n-2)+f(n-1)$. 计算 $f(20)$的值.

解 ☞

```
Clear[f]
f[1] = 1;
f[2] = 2;
f[3] = 3;
f[n_] := f[n] = f[n - 3] + f[n - 2] + f[n - 1];
f[20]
101902
```

2.66 定义一个函数,它表示点(x, y)到点(3,4)之间的距离,并计算函数在点 (5, −2)的值.

解 ☞

```
f[x_, y_] = √((x - 3)^2 + (y - 4)^2); 或者
f[x_, y_] = Sqrt[(x - 3 )^2 + (y - 4)^2];
f[5, -2]
2 √10
```

2.67 定义一个函数,它表示点 $(x1, y1)$ 与点$(x2, y2)$ 之间的距离,并利用它计算点 (2, 3) 到点(8 ,11) 之间的距离.

解 ☞

```
d[x1_, y1_, x2_, y2_] = √((x2 - x1)^2 + (y2 - y1)^2); 或者
d[x1_, y1_, x2_, y2_] = Sqrt[(x2 - x1)^2 + (y2 - y1)^2];
d[2, 3, 8, 11]
10
```

2.68 边长分别为 a, b, c 的三角形的面积是由海伦(Heron)公式给出的:

$$K = \sqrt{s(s-a)(s-b)(s-c)},$$

其中 $s=\frac{a+b+c}{2}$. 把三角形的面积表示成 a, b, c 的函数,并利用它计算三角形边长为(a) 3, 4, 5; (b) 5, 9, 12 的面积.

解 ☞

```
s = (a + b + c)/2;
k[a_, b_, c_] = √(s(s - a)(s - b)(s - c));
k[3, 4, 5]
6
```

```
k[5, 9, 12]
4 √26
```

2.10 函数的运算

如果 f, g 为两个具有相同定义域 D 的函数，那么我们就可以逐点定义它们的和、差、积与商：

$$(f+g)(x)=f(x)+g(x) \quad 对 D 中所有的 x,$$
$$(f-g)(x)=f(x)-g(x) \quad 对 D 中所有的 x,$$
$$(fg)(x)=f(x)g(x) \quad 对 D 中所有的 x,$$
$$(f/g)(x)=f(x)/g(x) \quad 对 D 中所有满足 g(x)\neq 0 的 x.$$

如果 x 在 g 的定义域内，而 $g(x)$ 在 f 的定义域内，那么可以定义复合函数 $f\circ g$：

$$f\circ g(x) = f(g(x)).$$

类似地可以定义函数 $g\circ f$.

下述例子说明了如何构造这些函数的方法.

例 67
```
f[x_] = √x;
g[x_] = x^2 + 2x + 3;
h1[x_] = f[x] + g[x]
3 + √x + 2x + x^2
h2[x_] = f[x] - g[x]
-3 + √x - 2x - x^2
h3[x_] = f[x]g[x]
√x(3 + 2x + x^2)
h4[x_] = f[x]/g[x]
√x/(3 + 2x + x^2)
h5[x_] = f[g[x]]
√(3 + 2x + x^2)
h6[x_] = g[f[x]]
3 + 2√x + x
```

两个或者更多函数的复合可以用 **Composition** 命令实现. 注意 Composition 是一个函数运算，因此它的参数必须是函数 f，而不是 f[x].

■ **Composition[f1, f2, f3, …]** 构造函数复合 f1 o f2 o f3….

例 68
```
f[x_] = √x;
g[x_] = x^2 + 2x + 3;
h1 = Composition[f, g];
h1[x]
√(3 + 2x + x^2)
h2 = composition[g, f];
h2[x]
3 + 2√x + x
```

如果要计算函数与自身的复合，那么当然可以用 Composition[f, f], Composition[f, f, f],等等.而更方便的工具是 Nest 与 NestList.

- **Nest[f, 表达式, n]** 重复应用 f 到表达式上 n 次.
- **NestList[f, 表达式, n]** 重复应用 f 到表达式上 n 次,并返回一个由所有中间计算结果组成列表.(列表在第三章中详加讨论.)

例 69
```
f[x_] = x^2;
Nest[f, x, 5]
```
x^{32}
```
NestList[f, x, 5]
```
$\{x, x^2, x^4, x^8, x^{16}, x^{32}\}$
```
Nest[f, 2x + 3, 5]
```
$(3+2x)^{32}$
```
NestList[f, 2x + 3, 5]
```
$\{3+2x, (3+2x)^2, (3+2x)^4, (3+2x)^8, (3+2x)^{16}, (3+2x)^{32}\}$

习题解答

2.69 如果 $f(x)=\sin x+2\cos x$, $g(x)=2\sin x-3\cos x$，构造函数$(f+g)(x)$, $(f-g)(x)$, $(fg)(x)$和$(f/g)(x)$,并计算它们在 $\pi/2$ 的值.

解 ☞

```
f[x_] = Sin[x] + 2 Cos[x];
g[x_] = 2 Sin[x] - 3 Cos[x];
h1[x_] = f[x] + g[x]
-Cos[x] + 3 Sin[x]
h2[x_] = f[x] - g[x]
5 Cos[x] - Sin[x]
h3[x_] = f[x]g[x]
(2 Cos[x] + Sin[x])(-3 Cos[x] + 2 Sin[x])
h4[x_] = f[x]/g[x]
```
$\frac{2\,\text{Cos}[x]+\text{Sin}[x]}{-3\text{Cos}[x]+2\text{Sin}[x]}$
```
h1[π/2]
3
h2[π/2]
-1
h3[π/2]
2
h4[π/2]
```
$\frac{1}{2}$

2.70 令 $f(x)=\sqrt{1+x}$,计算$(f\circ f\circ f\circ f\circ f)(x)$.

解 ☞

$\texttt{f[x_]} = \sqrt{1+x}$;

```
Nest[f, x, 5]
```

$\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+x}}}}}$

2.71 令 $f(x)=\frac{1}{1+x}$. 把 f 重复应用到 x 上五次，并显示每次的结果. 然后把 f 应用到 $x=1$ 上 20 次，得到了什么结果？

解 ☞

$f[x_]=\frac{1}{1+x}$

```
NestList[f, x, 5]
```

$\left\{x,\frac{1}{1+x},\frac{1}{1+\frac{1}{1+x}},\frac{1}{1+\frac{1}{1+\frac{1}{1+x}}},\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+x}}}},\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+x}}}}}\right\}$

```
NestList[f, 1, 20]
```

$\left\{1,\frac{1}{2},\frac{2}{3},\frac{3}{5},\frac{5}{8},\frac{8}{13},\frac{13}{21},\frac{21}{34},\frac{34}{55},\frac{55}{89},\frac{89}{144},\frac{144}{233},\frac{233}{377},\frac{377}{610},\frac{610}{987},\frac{987}{1597},\frac{1597}{2584},\frac{2584}{4181},\frac{4181}{6765},\frac{6765}{10946},\frac{10946}{17711}\right\}$

```
% //N
{1., 0.5, 0.666667, 0.6, 0.625, 0.615385, 0.619048, 0.617647,
0.618182, 0.617978, 0.618056, 0.618026, 0.618037, 0.618033,
0.618034, 0.618034, 0.618034, 0.618034, 0.618034, 0.618034,
0.618034}
```

得到的数均在接近 0.618034 的近似值的范围内.

2.72 如果 x 为 $\sqrt{a}$ 的近似值，那么可以证明 $\frac{1}{2}\left(x+\frac{a}{x}\right)$ 为更好的近似值.（这是牛顿方法的一个特殊情形.）利用 **NestList** 观察从 $x=100$ 开始而得到的 $\sqrt{3}$ 的前 10 个近似值.

解 ☞

```
a = 3;
```

$f[x_]=\frac{1}{2}\left(x+\frac{a}{x}\right);$

```
NestList[f, 100, 10] //N
{100., 50.015, 25.0375, 12.5787, 6.40858, 3.43835, 2.15543, 1.77363, 1.73254,
1.73205, 1.73205}
```

2.11 模块

在缺省情况下，Mathematica 假定所有的对象（例如变量）都是全局性的. 这就是说，例如，如果你定义了变量 x，其值为 3，那么 x 的值一直为 3，除非你进行了修改. 与之相反，局部对象只有在一组特定指令内才有一定的作用.

通过模块可以定义局部变量，局部变量只在模块内有定义. 在模块外面，对象可能就没有定义，或者具有完全不同的值.

■ **Module[{变量 1, 变量 2,…}, 主体]** 定义拥有局部变量变量 1,变量 2,…的模块.

■ **Module[{变量 1 = 值 1, 变量 2 = 值 2,…}, 主体]** 定义拥有局部变量变量1,变量 2,…的模块,并且这里每个变量具有指定的初始值.

例 70

```
x = 3;                        ←全局变量 x 置为 3.
Module[{x = 8}, x + 1]        ←定义的模块内局部变量 x 初始值为 8
9                             ←x 被增加.
x                             ← 调用的是全局变量 x.
3                             ←返回的是 x 的原始值.
```

有时候把几条命令组织成一个整体,从而可以当作一个组来执行,这在实际中是非常有用的.特别在复杂结构中包含循环时尤其如此.在主体中可以包含几条命令,只要命令之间用分号分开就可以了.

例 71

```
Module[{x = 1, y = 2}, x = x + 3; y = y + 4; Print[x y]]
24            ← x 与 y 分别具有初始值 1 与 2,然后 x 增加 3,y 增加 4,再把它们乘起来.
```

有时在定义函数时,把它的值设计为一个模块,在应用时就非常方便.当处理的函数的定义相当复杂时,这可以使我们具有相当大的弹性.当用这种方式定义函数时,必须用延时赋值符号 :=.

例 72 下面的语句定义了阶乘函数.这里假定 x0 的值为非负整数.变量 fact 与 x 分别被初始化为 1 和 x0,都是局部变量,因此与程序中其他地方的同名变量不会发生冲突.这里 x0 为哑变量.

```
f[x0_] := Module[{fact = 1, x = x0},
                  While[x>1, fact = x * fact; x = x - 1];
Print[fact]]
f[0]
1
f[5]
120
f[10]
3628800
```

为了更清楚地了解模块的工作机理,我们考虑下面这个例子.虽然同样的模块被执行了三次,变量都是 x,但它实际上被赋予三个不同的局部名称.正是由于这种聪明的屏蔽措施,使得三个变量是相互独立的,从而不会与全局变量 x 冲突.

例 73

```
x = 3
3
Module[{x}, Print[x]]
x$5
Module[{x}, Print[x]]
x$6
Module[{x}, Print[x]]
x$7
x
3
```

正是由于三个值被赋予不同的内部名称,它们才不会发生冲突.

习题解答

2.73 编写一个模块,它取一个整数,返回它的所有的因子.

解 ☞

```
factorlist[x0_] :=
Module[{x = 1}, While[x ≤ x0, If[Mod[x0, x] == 0, Print[x]]; x++]]
```

```
factorlist[1]
1

factorlist[10]
1
2
5
10

factorlist[11]
1
11
```

```
factorlist[90]
1
2
3
5
6
9
10
15
18
30
45
90
```

2.74 确定一个素数在素数序列中所处位置的比较粗糙的方法就是查看由所有素数构成的列表,直到遇到或超过所考虑的数为止,从而可以确定它的位置.如果指定数不在列表中,就知道它不是素数.构造一个模块,利用上述方法来判断一个数是否为素数,如果它是素数,给出它的位置;如果它不是素数,返回一条信息说明它不是素数.

解 ☞

```
pos[x0_] := Module[{x = 1, prm},
  prm = False;
  While[Prime[x] ≤ x0 && Not[prm], If[Prime[x] == x0, prm = True]; x++];
If[prm, Print[x - 1], Print["Not a Prime"]]]
pos[1]
Not a Prime
pos[2]
1
pos[3]
2
pos[101]
26
pos[1001]
Not a Prime
```

2.75 在数学中有一个著名的猜想,那就是:如果从正整数 n 开始,当 n 为偶数时,就用 $n/2$ 代替它,当 n 为奇数时,就用 $3n+1$ 代替它,按这种方式进行多次迭代,那么总会得到 1.(这个猜测还没有被证明或否定.)构造一个模块,模拟上述迭代过程.

解 ☞

我们首先定义一个函数 successor，它定义一次迭代步骤.

```
successor[n_] := If[EvenQ[n], n/2, 3n + 1]
```

接下来，给出模块 allvalues，它给出从 n 开始的所有后续数的清单.

```
allvalues[n_] := Module[{m = n}, While[m≠1, m = successor[m]; Print[m]]]
allvalues[6]
3
10
5
16
8
4
2
1
```

由于当 n 很大时，这个清单可能变得很长，而我们只对最后的结果以及得到这个结果的迭代次数感兴趣，因此下面这个模块可能更恰当.

```
finalvalue[n_] :=
  Module[{m = n, k = 0}, While[m ! = 1, m = successor[m]; k++];
    Print["final value = ", m, ", #iterations =", k]]
```

finalvalue 列出处理的最终结果，以及得到这个结果的迭代次数.

```
finalvalue[6]
finalvalue = 1, # iterations = 8
finalvalue[100]
finalvalue = 1, # iterations = 25
finalvalue[1000]
finalvalue = 1, # iterations = 11
```

第三章　列　　表

3.1　简介

所谓列表,就是由一组其他对象构成的对象.在阅读本章时,你就会发现列表有相当广泛的应用.因此 Mathematica 提供了一大类列表操作命令.

包含在列表内的对象是用大括号{}包围起来的.另外,**List** 命令也可以定义一个列表.

■ **List[元素]**,其中元素就是用逗号分开的各个列表对象,这条命令等价于{**元素**}.

例 1　{1, 2, 3, 4}为数的列表.List[a, b, c, d]为未定义符号的列表.

```
List[a, b, c, d]
{a, b, c, d}                    ← List[a, b, c, d]等价于{a, b, c, d}.
```

可以赋给列表一个符号名称,从而方便以后对它进行引用.对列表进行的操作就是对列表中每个元素进行的操作.

例 2

```
lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
1/lst
```

$\left\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \frac{1}{9}, \frac{1}{10}\right\}$,

> 之所以使用符号 **lst**,而不用 **list**,就是为了避免与 Mathematica 符号 **List** 冲突.

lst^2

{1, 4, 9, 16, 25, 36, 49, 64, 81, 100}

lst^2

{1,4,9,16,25,36,49,64,81,100}

$\sqrt{lst}$

$\{1,\sqrt{2},\sqrt{3},2,\sqrt{5},\sqrt{6},\sqrt{7},2\sqrt{2},3, \sqrt{10}\}$

如果有两个或多个列表具有同样数目的元素,那么可以利用标准运算创建新列表.

例 3

```
lst1 = {1, 2, 3, 4, 5};
lst2 = {2, 3, 2, 3, 2};
lst1 + lst2
{3, 5, 5, 7, 7}
lst1 * lst2
{2, 6, 6, 12, 10}
lst/lst2
```

$\{\frac{1}{2}, \frac{2}{3}, \frac{3}{2}, \frac{4}{3}, \frac{5}{2}\}$

$lst1^{lst2}$

{1, 8, 9, 64, 25}

习 题 解 答

3.1　构造列表,由从 1 到 10 的整数的阶乘组成.

解 ☞

```
lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
lst!
{1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800}
```

3.2 构造一个列表，由以 2 为底、以前 10 个正整数为指数的幂组成.

解 ☞

```
lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
2^lst
{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024}
```

3.3 构造一个列表，其元素分别是前五个正整数的平方与立方和.

解 ☞

```
lst = {1, 2, 3, 4, 5};
lst^2 + lst^3或 lst^2 + lst^3
{2, 12, 36, 80, 150}
```

3.4 定义 lst1 = {1, 3, 5, 7, 9}和 lst2 = {2, 4, 6, 8, 10}. 构造一个列表，它的五个元素就是这两个列表的对应元素的乘积.

解 ☞

```
lst1 = {1, 3, 5, 7, 9};
lst2 = {2, 4, 6, 8, 10};
lst1 * lst2
{2, 12, 30, 56, 90}
```

3.2 生成列表

最常见的列表就是等距数构成的列表. 利用 **Range** 命令可方便地构造这种列表. 在下面的语法描述中，m, n 和 d 不必为整数值，也可以是负值.

- **Range[n]** 生成由前 n 个相邻整数组成的列表.
- **Range[m, n]** 生成由从 m 到 n 的相邻整数组成的列表.
- **Range[m, n, d]** 生成由从 m 到 n 间隔为 d 的整数组成的列表.

例 4 **Range[10]**

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Range[5, 10]
{5, 6, 7, 8, 9, 10}
Range[25, 5, -2]
{25, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5}
Range[1/3, 1, 1/12]
```

$\left\{\frac{1}{3}, \frac{5}{12}, \frac{1}{2}, \frac{7}{12}, \frac{2}{3}, \frac{3}{4}, \frac{5}{6}, \frac{11}{12}, 1\right\}$

```
Range[1, 2, .1]
{1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.}
```

具有更复杂结构的列表可以用 **Table** 和 **Array** 命令创建.这两条命令具有几种不同的形式.

- **Table[表达式, {n}]** 生成一个列表,由表达式的 n 份拷贝组成.
- **Table[表达式, {k, n}]** 生成一个列表.由表达式在 k 从 1 变化到 n 时的值组成.
- **Table[表达式, {k, m, n}]** 生成一个列表,由表达式在 k 从 m 变化到 n 时的值组成.
- **Table[表达式, {k, m, n, d}]** 生成一个列表,由表达式在 k 从 m 变化到 n 步长为 d 时的值组成.

例 5 **Table[x, {10}]**

{x, x, x, x, x, x, x, x, x, x}

Table[k^2, {k, 10}]

{1, 4, 9, 16, 25, 36, 49, 64, 81, 100}

Table[1/k, {k, 5, 13}]

$\left\{\frac{1}{5},\frac{1}{6},\frac{1}{7},\frac{1}{8},\frac{1}{9},\frac{1}{10},\frac{1}{11},\frac{1}{12},\frac{1}{13}\right\}$

Table[$\sqrt{k}$, {k, 5, 13, 2}]

$\{\sqrt{5},\sqrt{7},3,\sqrt{11},\sqrt{13}\}$

- **Array[f, n]** 生成一个列表,由 n 个值 f[1], f[2], …, f[n]组成.
- **Array[f, n, r]** 生成一个列表,由 i 从 r 开始的 n 个值 f[i]组成.

例 6 **f[x_] = x^2 + x + 1;**

Array[f, 10]

{3, 7, 13, 21, 31, 43, 57, 73, 91, 111}

Array[f, 10, 0]

{1, 3, 7, 13, 21, 31, 43, 57, 73, 91}　　←第一个元素是 f[0].

嵌套列表就是在列表中还包含其他的列表.例如,

{{1, 2, 3, 4}, {2, 3, 4, 5}, {3, 4, 5, 6}}

就是由三个元素组成的嵌套列表,其中每个元素都是由四个整数构成的列表.可以用 Table 和 Array 命令生成嵌套列表.

- **Table[表达式, {i, m_i, n_i}, {j, m_j, n_j}]** 生成一个嵌套列表,它由表达式在 j 从 m_j 变到 n_j, i 从 m_i 变到 n_i 时的值组成.
- **Array[f, {m, n}]** 生成一个嵌套列表,它是由 m 个元素构成的数组,而其中的每个元素都是由 n 个元素构成的数组,它们的值为 f[i, j],这里 j 从 1 变到 n, i 从 1 变到 m,其中 f 为二元函数.
- **Array[f, {m, n}, {r, s}]** 生成一个嵌套列表,它是由 m 个元素构成的数组,而其中每个元素是由 n 个元素构成的数组.其中第一个子列表的第一个元素是f[r, s].

在上面描述的语法中,最右边的指标变化得最快.对应于这个指标的列表是最内部的列表.所有的指标都有单位增长.

上面给出的语法描述,都可以自然扩展为有更多层次的情形.

例 7 **Table[i + j, {i, 1, 3}, {j, 1, 5}]**

{{2, 3, 4, 5, 6}, {3, 4, 5, 6, 7}, {4, 5, 6, 7, 8}}

对应于 i 的每个值,生成列表{i + 1, i + 2, i + 3, i + 4, i + 5}.这样得到的三个列表被

包围在最外层的大括号内.

```
Table[i + j, {i, 1, 5}, {j, 1, 3}]
{{2, 3, 4}, {3, 4, 5}, {4, 5, 6}, {5, 6, 7}, {6, 7, 8}}
```

例 8

```
g[x_, y_] = x^2 + 3y;
Array[g, {3, 4}]
{{4, 7, 10, 13}, {7, 10, 13, 16}, {12, 15, 18, 21}}
Array[g, {4, 3}]
{{4, 7, 10}, {7, 10, 13}, {12, 15, 18}, {19, 22, 25}}
Array[g, {4, 3}, {0, 0}]
{{0, 3, 6}, {1, 4, 7}, {4, 7, 10}, {9, 12, 15}}
```

有时候构造字母和其他符号的列表是非常方便的.

- **Characters[字符串]** 生成由字符串中字符组成的列表.
- **CharacterRange["字符 1","字符 2"]** 生成从字符 1 到字符 2 的所有字符构成的列表,这里的顺序是基于标准(美式英语字母表)ASCII 码的.

例 9

```
Characters["Mathematica"]
{M, a, t, h, e, m, a, t, i, c, a}
```

例 10

```
CharacterRange["a", "e"]
{a, b, c, d, e}
CharacterRange[" ", "~"]
{ , !, ", #, $, %, &, ', (, ), *, +, ,, -, ., /, 0, 1,
2, 3, 4, 5, 6, 7, 8, 9, :, ;, <, =, >, ?, @, A, B, C, D,
E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X,
Y, Z, [, \, ], ^, _, `, a, b, c, d, e, f, g, h, i, j, k,
l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, {, |, }, ~}
```

虽然 Characters 和 CharacterRange 的输出好像是单个字符,但实际上它们是长度为 1 的字符串.按照 Mathematica 的约定,这里并不显示双引号.

例 11

```
digits = CharacterRange["0","9"]
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}          ←这些不是数字,而是长度为 1 的字符串.
FullForm[digits]
List["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]
```

习 题 解 答

3.5 构造一个列表,由所有为 7 的倍数,但不超过 100 的正整数组成.

解 ☞

Range[7, 100, 7] 或者 **Table[7k, {k, 1, 14}]**

```
{7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98}
```

3.6 构造一个列表,由前 10 个素数组成.

解 ☞

```
Table[Prime[k], {k, 1, 10}]
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
```

解 ☞

```
Array[Prime, 10]
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
```

解 ☞

```
Prime[Range[10]]
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29}
```

3.7 构造一个列表,由前 10 个偶数的倒数组成.

解 ☞

`Table[1/k, {k, 2, 20, 2}]` 或者 `1/Range[2, 20, 2]`

$\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \frac{1}{10}, \frac{1}{12}, \frac{1}{14}, \frac{1}{16}, \frac{1}{18}, \frac{1}{20}\right\}$

3.8 构造一个有五个对象的列表,其中每个对象是由六个整数组成的列表.第一个列表是由 2 的前 6 个倍数组成,第二个是 3 的倍数,第三个是 4 的倍数,依此类推.

解 ☞

```
Table[i * j, {i, 2, 6}, {j, 1, 6}]
{{2, 4, 6, 8, 10, 12}, {3, 6, 9, 12, 15, 18}, {4, 8, 12, 16, 20, 24}, {5, 10, 15,
20, 25, 30}, {6, 12, 18, 24, 30, 36}}
```

3.9 令 $p(x) = x^2 - 8x + 10$. 计算 $p(x)$ 在 $x = 1, 2, 3, \cdots, 10$ 时的值.

解 ☞

```
p[x_] = x^2 - 8x + 10;
Array[p, 10] 或者 p[Range[10]]
{3, -2, -5, -6, -5, -2, 3, 10, 19, 30}
```

3.3 列表的操作

- `Length[lst]` 给出 lst 的长度,即在 lst 中元素的数目.
- `First[lst]` 给出 lst 中位于第一个位置上的元素.
- `Last[lst]` 给出 lst 中位于最后位置上的元素.

例 12

```
lst = {a, b, c, d, e, f, g};
Length[lst]
7
First[lst]
a
Last[lst]
g
```

函数 `Part` 返回列表中的某个元素.

■ **Part[lst, k]** 或 **lst[[k]]** 返回 lst 中的第 k 个元素.

■ **Part[lst, -k]** 或 **lst[[-k]]** 返回 lst 中倒数第 k 个元素.

Part[lst, 1]与**Part[lst, -1]**分别等价于**First[lst]**与 **Last[lst]**.

例 13 **lst = {a, b, c, d, e, f, g};**

```
Part[lst, 1] 或者 lst[[1]]
a
Part[lst, 3] 或者 lst[[3]]
c
Part[lst, -3] 或者 lst[[-3]]
e
Part[lst, -1] 或者 lst[[-1]]
g
```

例 14 **lst = {{a, b, c, d}, {e, f, g, h}, {i, j, k, l}};**

```
First[lst]
{a, b, c, d}
Last[lst]
{i, j, k, l}
lst[[2]]
{e, f, g, h}
```

由于 lst[[2]]本身就是一个列表,因此它的第三个元素就可以用 lst[[2]][[3]](第 2 个表的第 3 个元素)得到. 为了方便起见,这也可以表示为 lst[[2,3]]或者 Part[lst, 2, 3]. 显然还可以用 Part[Part[lst, 2], 3],但这实在太不方便了.

例 15 **lst = {{a, b, c, d}, {e, f, g, h}, {i, j, k, l}};**

```
lst[[2]][[3]]
g
lst[[2, 3]]
g
Part[lst, 2, 3]
g
Part[Part[lst, 2], 3]
g
```

有几种不同的方法修改列表. 在下面的语法中, lst 为任意列表.

■ **Rest[lst]** 返回删除 lst 中第一个对象后的结果.

■ **Take[lst, n]** 返回一个列表, 由 lst 的前 n 个对象组成.

■ **Take[lst, {n}]** 返回一个列表, 由 lst 的第 n 个对象组成.

■ **Take[lst, -n]** 返回一个列表, 由 lst 的后 n 个对象组成.

■ **Take[lst, {-n}]** 返回一个列表, 由 lst 的倒数第 n 个对象组成.

■ **Take[lst, {m, n}]** 返回一个列表, 由 lst 中从第 m 个到第 n 个对象组成.

例 16 **lst = {a, b, c, d, e, f, g};**

```
Rest[lst]
{b, c, d, e, f, g}
Take[lst, 3]
{a, b, c}
Take[lst, -3]
{e, f, g}
Take[lst, {3}]
{c}
Take[lst, {-3}]
{e}
Take[lst, {2, 5}]
{b, c, d, e}
```

可以用 **Delete** 命令从列表中删除元素.

- **Delete[lst, n]** 删除在 lst 中第 n 个位置上的元素.
- **Delete[lst, -n]** 删除在 lst 中倒数第 n 个位置上的元素.

例 17

```
lst = {a, b, c, d, e, f, g};
Delete[lst, 3]
{a, b, d, e, f, g}
Delete[lst, -3]
{a, b, c, d, f, g}
```

相比起来, **Drop** 函数就富有弹性.

- **Drop[lst, n]** 返回 lst, 其前 n 个对象被删除.
- **Drop[lst, -n]** 返回 lst, 其后 n 个对象被删除.
- **Drop[lst, {n}]** 返回 lst, 其第 n 个对象被删除.
- **Drop[lst, {-n}]** 返回 lst, 其倒数第 n 个对象被删除.
- **Drop[lst, {m, n}]** 返回 lst, 其从第 m 个到第 n 个对象被删除.

注意　**Drop[lst, {n}]** 等价于 **Delete[lst, n]**, **Drop[lst, {-n}]** 等价于 **Delete[lst, -n]**.

例 18

```
lst = {a, b, c, d, e, f, g};
Drop[lst, 2]
{c, d, e, f, g}
Drop[lst, -2]
{a, b, c, d, e}
Drop[lst, {2}]
{a, c, d, e, f, g}
Drop[lst, {-2}]
{a, b, c, d, e, g}
Drop[lst, {2, 4}]
{a, e, f, g}
```

有许多列表操作可以向列表中插入元素.

- **Append[lst, x]** 把 x 插入到最后一个元素的右边,并返回 lst.
- **Prepend[lst, x]** 把 x 插入到第一个元素的左边,并返回 lst.
- **Insert[lst, x, n]** 把 x 插入到第 n 个位置上,并返回 lst.
- **Insert[lst, x, -n]** 把 x 插入到倒数第 n 个位置上,并返回 lst.

例 19

```
lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
Append[lst, x]
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, x}
Prepend[lst, x]
{x, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Insert[lst, x, 4]
{1, 2, 3, x, 4, 5, 6, 7, 8, 9, 10}
Insert[lst, x, -4]
{1, 2, 3, 4, 5, 6, 7, x, 8, 9, 10}
```

可以用 **ReplacePart** 把列表中的对象用其他的对象替换.

- **ReplacePart[lst, x, n]** 把 lst 中第 n 个位置上的对象用 x 替换.
- **ReplacePart[lst, x, -n]** 把 lst 中倒数第 n 个位置上的对象用 x 替换.

例 20

```
lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
ReplacePart[lst, x, 7]
{1, 2, 3, 4, 5, 6, x, 8, 9, 10}
ReplacePart[lst, x, -7]
{1, 2, 3, x, 5, 6, 7, 8, 9, 10}
```

可以用 **Sort** 和 **Reverse** 命令重排列表.

- **Sort[lst]** 按升序对列表 lst 中的元素进行排序.其中实数是按它们的数值大小排序的.字母按字母表顺序进行,其中大写字母排在小写字母后面.
- **Reverse[lst]** 把 lst 中的元素顺序颠倒过来.

例 21

```
lst = {1, 5, -3, 0, 2.5};
Sort[lst]
{-3, 0, 1, 2.5, 5}
```

例 22

```
lst = {z, x, Y, w, X, y, Z, W};
Sort[lst]
{w, W, x, X, y, Y, z, Z}
```

例 23

```
lst = {a, b, c, d, e, f, g};
Reverse[lst]
{g, f, e, d, c, b, a}
```

利用 **RotateLeft** 和 **RotateRight**,可以循环变换列表中元素的位置.

- **RotateLeft[lst]** lst 中每个元素都依次向左移动一个位置,而最左边的元素移到列

表的最右边.

- **RotateLeft[lst, n]** lst 中每个元素都依次向左移动 n 个位置,而最左边 n 个元素按同样的相对位置移到列表的最右边.如果 n 为负数,则向右循环.
- **RotateRight[lst]** lst 中每个元素都依次向右移动一个位置,而最右边的元素移到列表的最左边.
- **RotateRight[lst, n]** lst 中每个元素都依次向右移动 n 个位置,而最右边 n 个元素按同样的相对位置移到列表的最左边.如果 n 为负数,则向左循环.

例 24
```
lst = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
RotateLeft[lst]
{2, 3, 4, 5, 6, 7, 8, 9, 10, 1}
RotateLeft[lst, 3]
{4, 5, 6, 7, 8, 9, 10, 1, 2, 3}
RotateLeft[lst, -3]
{8, 9, 10, 1, 2, 3, 4, 5, 6, 7}
RotateRight[lst]
{10, 1, 2, 3, 4, 5, 6, 7, 8, 9}
RotateRight[lst, 3]
{8, 9, 10, 1, 2, 3, 4, 5, 6, 7}
RotateRight[lst, -3]
{4, 5, 6, 7, 8, 9, 10, 1, 2, 3}
```

列表可以用 **Join** 命令组合在一起.

- **Join[lst1, lst2]** 把两个列表 lst1,lst2 组合成一个列表,新列表由 lst1 和 lst2 中的元素组成.

Join 不会去掉重复元素.然而可以利用 Union 命令方便地消去重复元素(见第 5.4 节).Join 命令可以按自然的方法推广到组合多个列表的情形.

例 25
```
lst1 = {1, 2, 3, 4, 5 };
lst2 = {3, 4, 5, 6, 7 };
Join[lst1, lst2]
{1, 2, 3, 4, 5, 3, 4, 5, 6, 7}
```

嵌套列表是很常见的,它可以具有相当复杂的结构.有几条 Mathematica 命令可以帮助你理解与操作嵌套列表.

- **Depth[lst]** 返回在列表结构中层次数加 1 之后的结果.
- **Level[lst, {levelspec}]** 返回一个列表,它由 lst 中 levelspec 所指定层次上的元素构成.
- **Level[lst, levelspec]** 返回一个列表,它由 lst 中在 levelspec 所指定层次或者在该层次以下的元素构成.

例 26
```
lst = {1, {2, {3, 4, 5}}};
```

```
Depth[lst]
4
Level[lst, {1}]
{1, {2, {3, 4, 5}}}
Level[lst, {2}]
{2, {3, 4, 5}}
Level[lst, {3}]
{3, 4, 5}
Level[lst, 3]
{1, 2, 3, 4, 5, {3, 4, 5}, {2, {3, 4, 5}}}
```

4－1 = 3. 这个结果说明 lst 中包含列表,而内部列表中还包含列表. 要注意的是 Depth 返回的结果总比列表中实际层数多一. 这是由于 Mathematica 处理列表的技术上的原因导致的. 在目前情形上,只要记住层数比 Depth 小 1 就可以了.

- **Flatten[lst]** 把嵌套列表转化为由包含在 lst 最内层的元素构成的简单列表.
- **Flatten[lst, n]** 对嵌套列表进行 n 次平整,每次都把最外面的层次去掉.
- **FlattenAt[lst, n]** 把列表中位于第 n 个位置上的子列表进行一层平整. 如果 n 为负数, Mathematica 倒过来计数,即从尾部开始.

例 27

```
lst = {1, {2, 3}, {4, 5, {6}}, {7, {8, {9, 10}}}};
Flatten[lst]
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
Flatten[lst, 1]
{1, 2, 3, 4, 5, {6}, 7, {8, {9, 10}}}
Flatten[lst, 2]
{1, 2, 3, 4, 5, 6, 7, 8, {9, 10}}
FlattenAt[lst, 3]
{1, {2, 3}, 4, 5, {6}, {7, {8, {9, 10}}}}     ← 只有第三号位置上的子列表被平整了一层.
FlattenAt[lst, -1]
{1, {2, 3}, {4, 5, {6}}, 7, {8, {9, 10 }}}
```

Flatten 把嵌套列表转化为更简单的列表. 而 **Partition** 则以一种有组织的而且方便的方法把简单的列表转化为嵌套列表.

- **Partition[lst, k]** 把 lst 转化为长度为 k 的子列表. 如果 lst 中包含 kn + m 个元素,其中 m<k, 那么 Partition 就会创建 n 个子列表,而余下的 m 个元素被去掉.

Partition 是一个生成表格与矩阵的方便的命令. 这里给出的只是最简单的命令形式. 如果读者有兴趣的话,可以从帮助索引中找到命令的其他形式.

例 28

```
lst = Range[12]
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
Partition[lst, 4]
{{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}}
Partition[lst, 5]
{{1, 2, 3, 4, 5}, {6, 7, 8, 9, 10}}
Partition[lst, 6]
{{1, 2, 3, 4, 5, 6}, {7, 8, 9, 10, 11, 12}}
```

习题解答

3.10 **Mathematica** 函数 **IntegerDigits** 返回包含在整数中的数字组成的列表. 利用这条命令, 计算包含在 100! 中共有多少位数字, 其中从左边和右边数第 50 位数字分别是多少?

解 ☞

```
lst = IntegerDigits[100!]
{9, 3, 3, 2, 6, 2, 1, 5, 4, 4, 3, 9, 4, 4, 1, 5, 2, 6, 8, 1, 6,
 9, 9, 2, 3, 8, 8, 5, 6, 2, 6, 6, 7, 0, 0, 4, 9, 0, 7, 1, 5, 9,
 6, 8, 2, 6, 4, 3, 8, 1, 6, 2, 1, 4, 6, 8, 5, 9, 2, 9, 6, 3, 8,
 9, 5, 2, 1, 7, 5, 9, 9, 9, 9, 3, 2, 2, 9, 9, 1, 5, 6, 0, 8, 9,
 4, 1, 4, 6, 3, 9, 7, 6, 1, 5, 6, 5, 1, 8, 2, 8, 6, 2, 5, 3, 6,
 9, 7, 9, 2, 0, 8, 2, 7, 2, 2, 3, 7, 5, 8, 2, 5, 1, 1, 8, 5, 2,
 1, 0, 9, 1, 6, 8, 6, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
Length[lst]
158
Part[lst, 50] 或者 lst[[50]]
1
Part[lst, -50] 或者 lst[[-50]]
2
```

3.11 计算第 100 个 Fibonacci 数的数字和.

解 ☞

应用 **IntegerDigits** 函数解决这个问题(见 3.10 题)

```
lst = IntegerDigits[Fibonacci[100]]
{3, 5, 4, 2, 2, 4, 8, 4, 8, 1, 7, 9, 2, 6, 1, 9, 1, 5, 0, 7, 5}
```

Sum[lst[[k]], {k, 1, Length[lst]}] 或者 $\sum_{k=1}^{\text{Length[lst]}}$ **lst [[k]]**

93

3.12 **Table[i * j, {i, 3, 10}, {j, 2, 7}]** 生成由数组成的嵌套列表. 把第 5 个子列表中的第 4 个数加到第 6 个子列表中的第 3 个数上.

解 ☞

```
Table[i * j, {i, 3, 10}, {j, 2, 7}]
{{6, 9, 12, 15, 18, 21}, {8, 12, 16, 20, 24, 28},
 {10, 15, 20, 25, 30, 35}, {12, 18, 24, 30, 36, 42},
 {14, 21, 28, 35, 42, 49}, {16, 24, 32, 40, 48, 56},
 {18, 27, 36, 45, 54, 63}, {20, 30, 40, 50, 60, 70}}
lst[[5, 4]] + lst[[6, 3]]
67
```

3.13 Mathematica 命令 **RealDigits** 返回由包含在一个近似实数的数字组成的列表. 这个列

表由两个子列表构成,第一个子列表由包含在近似值中的数字构成,第二个子列表由小数点左边的数字位数组成.计算 π 的有 20 位有效数字的近似值,并确定最后一位数前面的数字是多少.

解 ☞

```
approx = N[Pi, 20]
3.1415926535897932385
lst = RealDigits[approx]
{{3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2, 3, 8, 5}, 1}
lst[[1, -2]]
8
```

3.14 构造一个列表,由从 1 到 10 以及从 20 到 30 的相邻整数组成.

解 ☞

```
lst = Drop[Range[30], {11, 19}]
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30}
```

3.15 构造一个列表,由从 1 到 10 的整数,后接 99,再加上从 11 到 20 的整数组成.

解 ☞

```
lst = Insert[Range[20], 99, 11]
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 99, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

3.16 构造一个列表,由从 1 到 20 的整数按降序排列组成.

解 ☞

```
lst = Range[20, 1, -1]
{20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1}
```

解 ☞

```
lst = Range[20]//Reverse                    ← 这等价于 Reverse[Range[20]].
{20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1}
```

3.17 把单词 "MISSISSIPPI" 中的字母按字母顺序排列.

解 ☞

```
lst = Characters["MISSISSIPPI"]
{M, I, S, S, I, S, S, I, P, P, I}
Sort[lst]
{I, I, I, I, M, P, P, S, S, S, S}
```

3.18 构造一个列表,由从 0 到 2π,间隔为 $\pi/6$ 的数组成.

解 ☞

`Range[0, 2Pi, Pi/6]` 或者 `Range[0, 2π, π/6]`

$$\left\{0, \frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{5\pi}{6}, \pi, \frac{7\pi}{6}, \frac{4\pi}{3}, \frac{3\pi}{2}, \frac{5\pi}{3}, \frac{11\pi}{6}, 2\pi\right\}$$

3.19 Flavius Joseph 是公元 1 世纪的犹太历史学家.他记述了一个故事,说有 10 个犹太人住

在山洞里,为了不向罗马人投降,他们选择了一种一个接一个自杀的方式.具体方法就是他们围成一个圆圈,然后依次一个人杀掉与他相邻的人.那么谁是最后的生存者呢?

解 ☞

分别给这些人编号 1 到 10,并定义一个由这 10 个整数组成的列表.

```
lst = Rangle[10]
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

第一个死掉的人是 2 号.因此我们把它从列表中去掉,方法就是向左轮换列表,并把它从列表中删除.

```
lst = Rest[RotateLeft[lst]]
{3, 4, 5, 6, 7, 8, 9, 10, 1}
```

新的列表由 3 开始,编号 2 被去掉了.为了确定生存者,我们需要重复这个过程,直到只剩下一个编号.

```
lst = Rest[RotateLeft[lst]]
{5, 6, 7, 8, 9, 10, 1, 3}
lst = Rest[RotateLeft[lst]]
{7, 8, 9, 10, 1, 3, 5}
lst = Rest[RotateLeft[lst]]
{9, 10, 1, 3, 5, 7}
lst = Rest[RotateLeft[lst]]
{1, 3, 5, 7, 9}
lst = Rest[RotateLeft[lst]]
{5, 7, 9, 1}
lst = Rest[RotateLeft[lst]]
{9, 1, 5}
lst = Rest[RotateLeft[lst]]
{5, 9}
lst = Rest[RotateLeft[lst]]
{5}
```

5 号为生存者.

虽然一步一步地观察列表变化相当有趣,然而对于长的列表,上面的方法就不太适用了.更有效的方法是利用 While 循环.

```
lst = Range[10];
While[Length[lst] > 1, lst = Rest[RotateLeft[lst]]]
lst
{5}
```

3.20 确定在列表{a, {b, c}, {{d, e}, {f, g}, {{h, i}}, {j, {k, l, m}}}}中最高层次元素是什么.

解 ☞

```
lst = {a, {b, c}, {{d, e}, {f, g}, {{h, i}}, {j, {k, l, m}}}};
Depth[lst]
5
```

← 记住从这个数中减 1 才是最高层次.

```
Level[lst, {4}]
{h, i, k, l, m}
```

3.21 分别简化列表{a, {b, c}, {{d, e}, {f, g}, {{h, i}}, {j, {k, l, m}}}}一个层次和两个层次.

解 ☞

```
lst = {a, {b, c}, {{d, e}, {f, g}, {{h, i}}, {j, {k, l, m}}}};
Flatten[lst, 1]
{a, b, c, {d, e}, {f, g}, {{h, i}}, {j, {k, l, m}}}
Flatten[lst, 2]
{a, b, c, d, e, f, g, {h, i}, j, {k, l, m}}
```

3.22 从由 A 到 X 的字母构成的列表出发,构造出有 6 个子列表的列表,每个子列表中有四个字母.

解 ☞

```
lst = CharacterRange["A","X"]
{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X}
Partition[lst, 4]
{{A, B, C, D}, {E, F, G, H}, {I, J, K, L},
 {M, N, O, P}, {Q, R, S, T}, {U, V, W, X}}
```

3.4 集合论

集合在 Mathematica 中是用列表表示的. 可以用基本的列表操作函数 **Union**, **Intersection** 与 **Complement** 对其进行处理.

- **Union[lst1, lst2]** 把两个列表 lst1 与 lst2 组合成一个列表,并且去掉重复元素. 虽然在这里只是给出了两个列表,实际上可以出现任意数目的列表参数. 特别地,**Union[lst]**就会去掉 lst 中的重复元素.
- **Intersection[lst1, lst2]** 返回同时出现在 lst1 与 lst2 中元素的有序列表. 如果 lst1 与 lst2 没有交集,即没有共同元素,那么命令 **Intersection [lst1, lst2]**返回空列表{}.
- **Complement[universe, lst1]** 返回由包含在 universe 中,但不在 lst1 中元素组成的有序列表. 在这里 universe 代表全集.
- **Complement[universe, lst1, lst2]** 返回由包含在 universe 中,但不在 lst1 和 lst2 中元素构成的有序列表. 这条命令可以自然推广到多于两个列表的情形.

例 29

```
lst = {a, b, c, a, c, c, b, b}
{a, b, c, a, c, c, b, b}
Union[lst]
{a, b, c}
```

例 30

```
universe = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
lst1 = {1, 3, 5, 7};
lst2 = {5, 7, 8, 10};
Union[lst1, lst2]
```

```
{1, 3, 5, 7, 8, 10}
Intersection[lst1, lst2]
{5 ,7}
Complement[universe, lst1]
{2, 4, 6, 8, 9, 10}
Complement[universe, lst1, lst2]
{2, 4, 6, 9}
```

在 **BasicInput** 模板上的 ⋃ 与 ⋂ 符号也分别表示并集与交集.

- **lst1 ⋃lst2** 等价于 **Union [lst1, lst2]**.
- **lst1 ⋂lst2** 等价于 **Intersection [lst1, lst2]**.

例 31
```
lst1 = {1, 2, 3, 4, 5};
lst2 = {3, 4, 5, 6, 7};
lst1 ⋃lst2
{1, 2, 3, 4, 5, 6, 7}
lst1 ⋂lst2
{3, 4, 5}
```

在软件包 **DiscreteMath`Combinatorica`** 中有大量实用的与集合有关的命令.其中包含 **CartesianProduct**, **Subsets** 与 **KSubsets**.

由定义,两个集合 A 与 B 的笛卡儿积就是元素的有序对集合,其中在每对元素中,第一个元素来自于 A,第二个元素来自于 B.

- **CartesianProduct [lst1, lst2]** 返回 lst1 与 lst2 的笛卡儿积.

例 32
```
<<DiscreteMath`Combinatorica`          ← 这上载了相应软件包,见第一章.
lst1 = {a, b, c, d};
lst2 = {x, y, z};
CartesianProduct[lst1, lst2]
{{a, x}, {a, y}, {a, z}, {b, x}, {b, y}, {b, z},
 {c, x}, {c, y}, {c, z}, {d, x}, {d, y}, {d, z}}
```

所谓集合 A 的子集就是每个元素都是 A 中元素的集合.空集是每个集合的子集.包含空集在内,一个有 n 个元素的集合共有 2^n 个子集.由 A 的所有子集构成的集合称为 A 的幂集.

- **Subsets[lst]**返回一个列表,由 lst 的所有子集构成,其中包含空集,这也就是 lst 的幂集.
- **KSubsets [lst, k]** 返回一个列表,由 lst 的所有包含 k 个元素的子集构成.

例 33
```
<<DiscreteMath`Combinatorica`          ←如果已经上载了这个软件包,可以忽略.
lst = {a, b, c, d};
Subsets[lst]
{{}, {a}, {a, b}, {b}, {b, c}, {a, b, c}, {a, c}, {c}, {c, d},
 {a, c, d}, {a, b, c, d}, {b, c, d}, {b, d}, {a, b, d}, {a,d},
 {d}}
    Subsets[lst]//Sort                 ←Sort 使子集以更有组织的形式显示出来.
```

```
{{}, {a}, {b}, {c}, {d}, {a, b}, {a, c}, {a, d}, {b, c}, {b, d},
 {c, d}, {a, b, c}, {a, b, d}, {a, c, d}, {b, c, d}, {a, b, c, d}}
KSubsets[lst, 3]
{{a, b, c}, {a, b, d}, {a, c, d}, {b, c, d}}
```

习 题 解 答

3.23 在单词 "MISSISSIPPI" 中有哪些不同的字母?

解 ☞

```
Union[Characters["MISSISSIPPI"]]
{I, M, P, S}
```

3.24 计算集合{a, b, c, d, e, f, g}, {c, d, e, f, g, h, i} 与 {e, f, g, h, i, j, k} 的并集与交集.

解 ☞

```
set1 = {a, b, c, d, e, f, g};
set2 = {c, d, e, f, g, h, i};
set3 = {e, f, g, h, i, j, k};
Union[set1, set2, set3] 或者 set1 ⋃set2 ⋃set3
{a, b, c, d, e, f, g, h, i, j, k}
Intersection[set1, set2, set3] 或者set1 ⋂set2 ⋂set3
{e, f, g}
```

3.25 求出所有在集合 {a, b, c, d, e, f, g} 中但不在 {a, c, d, e} 中的元素.

解 ☞

```
set1 = {a, b, c, d, e, f, g};
set2 = {a, c, d, e };
Complement[set1, set2]
{b, f, g}
```

3.26 已知第20个素数为71,求出所有小于71但不是素数的数.

解 ☞

```
universe = Range[71];
primenumbers = Table[Prime[k], {k, 1, 20}];
Complement[universe, primenumbers]
{1, 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26,
 27, 28, 30, 32, 33, 34, 35, 36, 38, 39, 40, 42, 44, 45, 46, 48,
 49, 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68,
 69, 70}
```

3.27 构造一个列表,由所有的辅音字母组成.

解 ☞

```
letters =CharacterRange["a","z"];
vowels =Characters["aeiou"];
consonants =Complement[letters, vowels]
{b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z}
```

3.28 给出所有的小于 1 000，而且既是素数，也是 Fibonacci 数的数.

解 ☞

```
k = 1; lst1 = {};
While[Fibonacci[k]≤1000,
      lst1 = Append[lst, Fibonacci[k]]; k ++ ]
k = 1; lst2 = {};
While[Prime[k] ≤1000,
      lst2 = Append[lst2, Prime[k]]; k ++ ]
lst1 ∩lst2
{2, 3, 5, 13, 89, 233}
```

3.29 创建一个列表，由 {a, b, c, d, e} 的所有子集构成.共有多少个子集?

解 ☞

```
<<DiscreteMath`Combinatorica`                ←首先必须上载这个软件包.
letters = {a, b, c, d, e};
Subsets[letters]//Sort
{{}, {a}, {b}, {c}, {d}, {e}, {a, b}, {a, c}, {a, d}, {a, e},
 {b, c}, {b, d}, {b, e}, {c, d}, {c, e}, {d, e}, {a, b, c},
 {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d},
 {b, c, e}, {b, d, e}, {c, d, e}, {a, b, c, d}, {a, b, c, e},
 {a, b, d, e}, {a, c, d, e}, {b, c, d, e}, {a, b, c, d, e}}
Length[%]
32
```

3.30 创建一个列表，由 {a, b, c, d, e} 的所有包含三个元素的子集构成.有多少个这样的子集呢?

解 ☞

```
<<DiscreteMath`Combinatorica`
letters = {a, b, c, d, e};
KSubsets[letters, 3]
{{a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e},
 {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}}
```

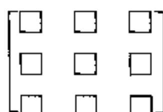
3.5 表格与矩阵

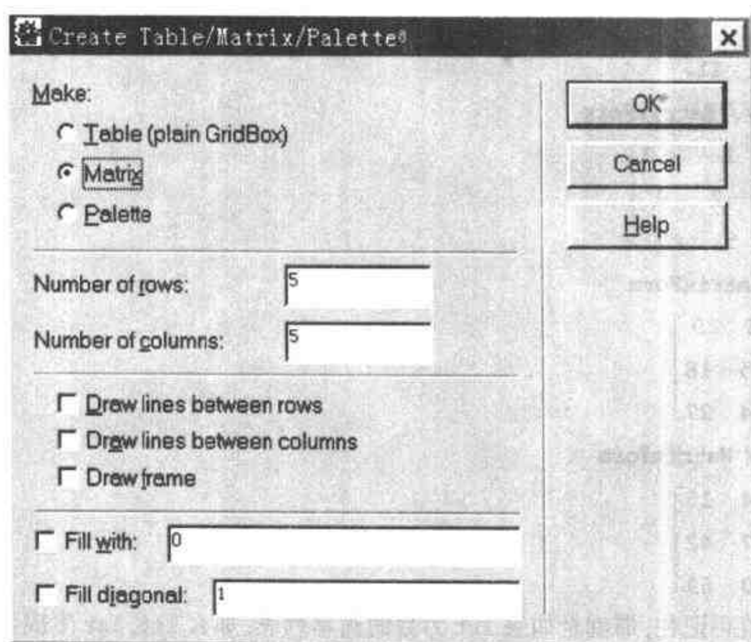
在 Mathematica 中用嵌套列表表示表格与矩阵.在 Mathematica 内部，它们的存贮方法没有区别，但如果使用 TableForm 与 MatrixForm 函数可以以不同的形式显示出结果.通常比较方便的用法是在矩阵名称的右边使用//TableForm 或//MatrixForm.

矩阵和表格可以按嵌套列表的方式直接输入.一个有 m 行 n 列的矩阵或表格,就是一个有 m 个子列表的嵌套列表,每个子列表包含 n 个元素.

也可以通过**Input ⇒Create Matrix/Table/Palette …** 便利地输入矩阵和表格.

这其中也有些非常方便的选项.点击 OK 就会产生空白格点,另外使用 [键] 键可以从一个元素进入取另一个元素中.





例 34 `lst = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}}`

`{{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}}`

`MatrixForm[lst]` 或者 **`lst//MatrixForm`**

$$\begin{bmatrix}1 & 2 & 3 & 4\\5 & 6 & 7 & 8\\9 & 10 & 11 & 12\end{bmatrix}$$

这里有两条特殊的生成矩阵的命令值得记住,因为在实际中经常用到它们.

■ **`IdentityMatrix[n]`** 生成 n×n 阶矩阵,其主对角线上元素为 1,其他元素全为 0.

■ **`DiagonalMatrix[lst]`** 生成一个阶为 `Length[lst]`的方阵,其对角线由 `lst` 中元素组成,而其他元素全是 0.这里要求 `lst` 为某种对象的列表.

例 35 **`IdentityMatrix[3]`**

$$\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}$$

`DiagonalMatrix[{1, 2, 3}]`

$$\begin{bmatrix}1 & 0 & 0\\0 & 2 & 0\\0 & 0 & 3\end{bmatrix}$$

矩阵一旦有了定义,就可以用加法、减法、常量乘法和矩阵乘法等运算进行矩阵组合.矩阵乘法运算用句号(.)表示.在第十二章线性代数中将对矩阵详加讨论.

> 这个矩阵是用 **Input ⇒Create Matrix/Table/Palette**…创建的，而 Mathematica 把矩阵显示为嵌套列表.

例 36　**A** = $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

```
{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
```

B = $\begin{bmatrix} 2 & 1 & 5 \\ 4 & 7 & 2 \\ 1 & 3 & 2 \end{bmatrix}$

```
{{2, 1, 5}, {4, 7, 2}, {1, 3, 2}}
```

A + B // MatrixForm

$\begin{bmatrix} 3 & 3 & 8 \\ 8 & 12 & 8 \\ 8 & 11 & 11 \end{bmatrix}$

A - B // MatrixForm

$\begin{bmatrix} -1 & 1 & -2 \\ 0 & -2 & 4 \\ 6 & 5 & 7 \end{bmatrix}$

3A // MatrixForm

$\begin{bmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \\ 21 & 24 & 27 \end{bmatrix}$

A . B // MatrixForm

$\begin{bmatrix} 13 & 24 & 15 \\ 34 & 57 & 42 \\ 25 & 90 & 69 \end{bmatrix}$

有种情形值得记住，那就是如果 lst 为数的简单列表，那么 lst.lst 生成元素的平方和.

例 37

```
lst = {1, 2, 3, 4, 5};
lst.lst
55
```

表格也是存贮为嵌套列表，但可以用 **TableForm** 显示成表格的形状. 虽然这条命令可以表示任何维的表格，但在本书中我们只讨论一维和二维表格.

■ **TableForm**[lst] 把 lst 中的元素以方阵形式显示出来.

如果没有指定其他选项的话，**lst//Table Form** 与 **TableForm[lst]** 是等价的.

例 38

```
lst = {{12, 7, 10}, {105, 205, 7}, {3, 30, 300}};
TableForm[lst] 或者 lst//TableForm
12      7       10
105     205     7
3       30      300
```

从上面的结果可见，在表格中的数字默认是左对齐的. 有时候这可能会使表格难以浏览. 可以用 **TableAlignments** 选项修正这一点.

TableAlignments 控制表格中数字的对齐方式.

- **TableAlignments →Left** 把列调整为左对齐(默认值).
- **TableAlignments →Right** 把列调整为右对齐.
- **TableAlignments →Center** 把列调整为居中对齐.

例 39
```
lst = {{12, 7, 10}, {105, 205, 7}, {3, 30, 300}};
TableForm[lst, TableAlignments ->Right]
12     7      10
105    205    7
3      30     300
TableForm[lst, TableAlignments ->Center]
12     7      10
105    205    7
3      30     300
```

在 TableForm 命令内,可以用 **TableHeadings** 选项插入行与列的标签.

- **TableHeadings →None** 为 Mathematica 的默认值.
- **TableHeadings →Automatic** 为行和列都生成由连续整数构成的标签.

例 40
```
lst = {{a, b, c}, {d, e, f}, {g, h, i}};
TableForm[lst, TableHeadings ->Automatic]
      1    2    3
1     a    b    c
2     d    e    f
3     g    h    i
```

每行和每列都可以用字符串(即包围在双引号内的字符)或 Mathematica 表达式作单独的标签.这个选项的一般形式为

- **TableHeadings → {行标签表,列标签表}**

其中行标签表就是行标签组成的列表,而列标签表就是列标签组成的列表.如果只希望有行标签,而列没有标签,或者与此相反,即只希望列有标签,而行没有标签,那么就有 **None** 取代行标签表或列标签表.

例 41
```
lst = {{a, b, c}, {d, e, f}, {g, h, i}};
TableForm[lst,
        TableHeadings ->{{"row1", "row2", "row3"},
        {"column1", "column2", "column3"}},
        TableAlignments ->Center]
        column1    column2    column3
row1       a          b          c
row2       d          e          f
row3       g          h          i
TableForm[lst,
    TableHeadings ->{None, {"column1", "column2", "column3"}},
```

```
   TableAlignments →Center]
column1      column2      column3
  a             b            c
  d             e            f
  g             h            i
TableForm[lst, TableHeadings →{{"row1", "row2", "row3"}, None},
          TableAlignments → Center]
row1     a     b     c
row2     d     e     f
row3     g     h     i
```

选项 **TableDirections** 确定表格中元素摆放的方式.

- **TableDirections →Column** (默认值)使得在显示表格时,每个内层列表的第一个元素显示在第一列上,然后每个内层列表的第二个元素显示在第二列上,依此类推.
- **TableDirections →Row** 在上面的描述中,用行代替列.

例 42

```
Clear[a]
lst = Array[a, {3, 4}]
{{a[1, 1], a[1, 2], a[1, 3], a[1, 4]}, {a[2, 1], a[2, 2], a[2, 3],
  a[2, 4]}, {a[3, 1], a[3, 2], a[3, 3], a[3, 4]}}
TableForm[lst, TableDirections →Column]
a[1, 1]  a[1, 2]  a[1, 3]  a[1, 4]
a[2, 1]  a[2, 2]  a[2, 3]  a[2, 4]
a[3, 1]  a[3, 2]  a[3, 3]  a[3, 4]
```

a[1, 1], a[2, 1]以及 a[3, 1]等元素构成第一列

```
TableForm[lst, TableDirections →Row]
a[1, 1]  a[2, 1]  a[3, 1]
a[1, 2]  a[2, 2]  a[3, 2]
a[1, 3]  a[2, 3]  a[3, 3]
a[1, 4]  a[2, 4]  a[3, 4]
```

a[1, 1], a[2, 1]以及 a[3, 1]等元素构成第一行.

缺省情况下,Mathematica 不会显示小数点右边的零.这有时候会导致表格的显示不一致.然而默认值可以被覆盖,从而按指定要求显示.

PaddedForm 对命令或计算的输出进行格式化处理.

- **PaddedForm [表达式, n]** 显示**表达式**的值,并保证结果占有 n 位数字的空白.命令的这一形式可以应用于整数或实数的近似值.

  注意　小数点并不算作一个位置.
- **PaddedForm[表达式, {n, f}]** 显示**表达式**的值,并保证结果占有 n 位数字的空白,而 f 表示小数点右边的位数.如果需要删除数字的话,就对数的小数部分进行四舍五入.

例 43

```
a = 123.456789;
PaddedForm[a, 12]
123.456789
```

←数的左边有二位数的空白.

```
PaddedForm[a, 20]
123.456789              ←数的左边有 11 位数的空白.
PaddedForm[a, {20, 3}]
123.457                 ←数的左边有 14 位数的空白,第三位小数被四舍五入为 7.
```

同其他函数一样,PaddedForm 函数可以作用到整个表格上.

例 44 下面这条命令定义了一个表格,由前 10 个正整数的平方根和立方根组成:

```
lst = Table[{n, N[√n], N[∛n]}, {n, 1, 10}];
```

首先用标准的 TableForm 命令显示表格.

```
TableForm[lst]
1     1.          1.
2     1.41421     1.25992
3     1.73205     1.44225
4     2.          1.5874
5     2.23607     1.70998
6     2.44949     1.81712
7     2.64575     1.91293
8     2.82843     2.
9     3.          2.08008
10    3.16228     2.15443
```

然后用 PaddedForm 规范整个表格.

```
PaddedForm[TableForm[lst],{7,5}]
1.00000      1.00000      1.00000
2.00000      1.41421      1.25992
3.00000      1.73205      1.44225
4.00000      2.00000      1.58740
5.00000      2.23607      1.70998
6.00000      2.44949      1.81712
7.00000      2.64575      1.91293
8.00000      2.82843      2.00000
9.00000      3.00000      2.08008
10.00000     3.16228      2.15443
```

如果希望对每一列进行不同的处理,那么就要对列表中的每个元素进行处理,而不能直接对整个表格进行操作.

```
paddedn := PaddedForm[n, 2]
paddedsquaroot := PaddedForm[N[√n], {11, 7}]
paddedcuberoot := PaddedForm[N[∛n], {7, 3}]
lst = Table[{paddedn, paddedsquaroot, paddedcuberoot},
          {n, 1, 10}];
TableForm[lst]
1      1.0000000      1.000
2      1.4142136      1.260
3      1.7320508      1.442
```

```
4      2.0000000      1.587
5      2.2360680      1.710
6      2.4494897      1.817
7      2.6457513      1.913
8      2.8284271      2.000
9      3.0000000      2.080
10     3.1622777      2.154
```

行与列之间的间距可以用 **TableSpacing** 调整.这个选项指定了添加到每个方向上的间距大小.

- **TableSpacing →{行间距, 列间距}**, 其中行间距指定在表格中相邻行之间的空行数,而列间距指定相邻列之间的空列数.

例 45

```
lst = {{a, b, c}, {d, e, f}, {g, h, i}};
TableForm[lst, TableSpacing →{0, 0}]
a  b  c          ←在行或列之间没有间隔.
d  e  f
g  h  i
TableForm[lst, TableSpacing →{1, 3}]
a  b  c          ← 行之间空一行,列之间空三列.
d  e  f
g  h  i
TableForm[lst, TableSpacing →{3, 1}]
a  b  c          ←行之间空三行,列之间空一列.
d  e  f
g  h  i
```

可以用 ColumnForm 把列表表示成单个列.

- **ColumnForm[lst]** 把 lst 表示成单个列.
- **ColumnForm[lst, 水平方向]** 这里要指定每行的对齐方式.对齐方式水平方向有 Left(表示左对齐,默认值),Center(右对齐)和 Right(居中).
- **ColumnForm[lst, 水平方向, 竖直方向]** 这里还可以指定每列的对齐方式.竖直方向对齐方式有 Above(向上对齐),Below(向下对齐,默认值)和 Center(居中).

例 46

```
lst ={a, bb, ccc};
ColumnForm[lst]
a
bb
ccc
ColumnForm[lst, Right]
a
bb
ccc
```

习 题 解 答

3.31 构造一个 3 阶矩阵,由相邻整数组成,向右向下增大.

解 ☞

```
lst = Table[3i + j, {i, 0, 2}, {j, 1, 3}]
{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
lst // MatrixForm
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

3.32 Hilbert 矩阵是一个方阵,其 (i, j)位置上的元素为$\frac{1}{i+j-1}$. 由此构造 5 阶 Hilbert 矩阵.

解 ☞

```
a[i_, j_] = 1/(i + j - 1);
hilbert = Array[a, {5, 5}]
```

$$\left\{\left\{1,\frac{1}{2},\frac{1}{3},\frac{1}{4},\frac{1}{5}\right\},\left\{\frac{1}{2},\frac{1}{3},\frac{1}{4},\frac{1}{5},\frac{1}{6}\right\},\left\{\frac{1}{3},\frac{1}{4},\frac{1}{5},\frac{1}{6},\frac{1}{7}\right\},\left\{\frac{1}{4},\frac{1}{5},\frac{1}{6},\frac{1}{7},\frac{1}{8}\right\},\left\{\frac{1}{5},\frac{1}{6},\frac{1}{7},\frac{1}{8},\frac{1}{9}\right\}\right\}$$

```
hilbert // MatrixForm
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

3.33 构造 5 阶单位阵.

解 ☞

```
IdentityMatrix[5]//MatrixForm
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.34 构造一个 5 阶矩阵,其对角元素为前 5 个素数,其余元素都是 0.

解 ☞

```
diag = Table[Prime[k], {k, 1, 5}]
```

{2, 3, 5, 7, 11}

```
DiagonalMatrix[diag] // MatrixForm
```

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 11 \end{bmatrix}$$

3.35 构造一个有三列的表格,第一列由从 1 到 10 的相邻整数构成,第二列和第三列则分别是它们的平方和立方.分别给这三列加上标签"integers"、"squares" 和 "cubes".

解 ☞

```
lst = Table[{k, k^2, k^3}, {k, 1, 10}];
TableForm[lst,
    TableHeadings →{None, {"integers", "squares", "cubes"}},
    TableAlignments →Right]
```

integers	squares	cubes
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

3.36 如果 c 表示以摄氏为单位的温度,那么它对应的华氏温度就是 $f=\frac{9}{5}c+32$. 构造一个列表,以水平显示的方式,说明从 1 到 10 摄氏度,每隔 1 度所等价的华氏温度.

解 ☞

$$f=\frac{9}{5}c+32$$

```
lst = Table[{c, PaddedForm[N[f], {3, 1}]}, {c, 1, 10}];
TableForm[lst, TableDirection →Row,
        TableHeadings →{None, {"Celsius", "Fahrenheit"}},
        TableAlignments →Center]
```

Celsius	1	2	3	4	5	6	7	8	9	10
Fahrenheit	33.8	35.6	37.4	39.2	41.0	42.8	44.6	46.4	48.2	50.0

3.37 构造一个列表,说明从 0°到 30°,每隔 5° 所对应的弧度.

解 ☞

```
lst = Table[{deg, N[deg Degree]}, {deg, 0, 30, 5}];
TableForm[lst, TableDirection → Row,
```

> **Degree** 为 Mathematica 常数(见第 二章).

```
   TableHeadings →{None, {"Degree", "Radians"}},
   TableAlignments →Center]
Degree    0    5          10          15          20          25          30
Radians   0.   0.0872665  0.174533    0.261799    0.309066    0.436332    0.523599
```

3.38 如果在银行帐户中有 p 美元,存期为 t 年,年利率为 r,每年支付 n 次,那么在 k 次支付后总钱数为 $p\left(1+\frac{r}{n}\right)^k$ 美元.若在帐号中存 1000 美元,利率为 6%,每年支付四次,列表说明在三年期中每次支付后的累加的钱数.

解 ☞

```
p = 1000;
r = .06;
n = 4;
t = 3;
a = p(1 + r/n)^k;
lst = Table[{k, a}, {k, 1, n t}];
TableForm[lst, TableHeadings →{None, {"period", "amount"}}]
```

period	amount
1	1015.
2	1030.22
3	1045.68
4	1061.36
5	1077.38
6	1093.44
7	1109.84
8	1126.49
9	1143.39
10	1160.54
11	1177.95
12	1195.62

3.39 如果在银行帐户中有 p 美元,年利率为 r,每年支付 n 次,那么在 t 年后总钱数为 $p\left(1+\frac{r}{n}\right)^{nt}$ 美元.如果利率是连续支付的,那么 t 年后总钱数为 pe^{rt}.若在帐号中存 1 000美元,年利率为 6%,列表说明若利率为每季度、每月、每天、以及连续支付时在 10 年内每个年终时累加的钱数.

解 ☞

```
p = 1000.;
r = .06;
a = PaddedForm[p (1 + r/4)^4t, {7, 2}];
b = PaddedForm[p (1 + r/12)^12t, {7, 2}];
c = PaddedForm[p (1 + r/365)^365t, {7, 2}];
d = PaddedForm[p Exp[r t], {7, 2}];
tt = PaddedForm[t, 2];
lst = Table[{tt, a, b, c, d}, {t, 1, 10}];
```

```
TableForm[lst, TableHeadings →{None, {"year", "quarterly",
      "monthly","daily","continuously"}}]
```

year	quarterly	monthly	daily	continuously
1	1061.36	1061.68	1061.83	1061.84
2	1126.49	1127.16	1127.49	1127.50
3	1195.62	1196.68	1197.20	1197.22
4	1268.99	1270.49	1271.22	1271.25
5	1346.86	1348.85	1349.83	1349.86
6	1429.50	1432.04	1433.29	1433.33
7	1517.22	1520.37	1521.91	1521.96
8	1601.32	1614.14	1616.01	1616.07
9	1709.14	1713.70	1715.93	1716.01
10	1814.02	1819.40	1822.03	1822.12

3.40 一笔价值 a 美元的抵押,每月支付 $\dfrac{a\times\frac{r}{12}}{1-\left(1+\frac{r}{12}\right)^{-12n}}$ 美元,其中 n 为借贷年限,r 每年的利率.构造一个列表,说明一笔 30 年的价值 250 000 美元的抵押,当利率从 6% 到 8%,间隔为 0.25% 时,每月所需要支付的钱数.

解 ☞

```
a = 250000;
n = 30;
payment := a (r/12) / (1 - (1 + r/12)^(-12n));
lst = Table[{PaddedForm[r, {5, 4}],
    PaddedForm[payment, {6, 2}]}, {r, .06, .08, .0025}];
TableForm[lst, TableHeadings →None, {"rate", "payment"}]]
```

rate	payment
0.0600	1498.88
0.0625	1539.29
0.0650	1580.17
0.0675	1621.50
0.0700	1663.26
0.0725	1705.44
0.0750	1748.04
0.0775	1791.03
0.0800	1834.41

第四章　二维图形

4.1　绘制一元函数的图形

如果你曾经利用过一种标准的程序设计语言绘制图形，那么就一定会非常欣赏 Mathematica 所提供的简单绘图方法．在许多情况中，只需要一行代码就可以生成具有两个变量的一个函数或更一般关系的图形演示．

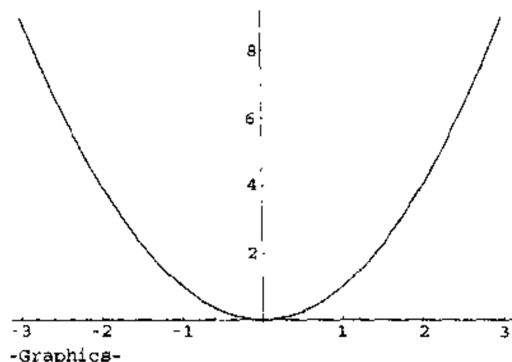
虽然在许多默认情形中，Mathematica 的方式是可以的，但还是需要有许多选项以控制某些细节．在本节我们将描述一些经常用到的选项，并给出许多例子，说明用这种方法绘制图形的简单性．

绘制函数图形的基本命令是 **Plot**．虽然在下面的描述中，用 x 作为独立变量，但实际上此处可以用任何符号．

■　**Plot[f[x], {x, xmin, xmax}]**绘制函数 $f(x)$在区间 xmin≤x≤xmax 上的图形．

例 1　绘制抛物线 $f(x)=x^2$ 从 －3 到 3 之间的图形．

```
Plot[x^2,{x, - 3,3}]
```



-Graphics-

这里跟在图形后面的描述（－Graphics－）说明该图形为"图形"（graphics）对象．可以在 Plot 命令的右边加上分号（;）禁止显示它．

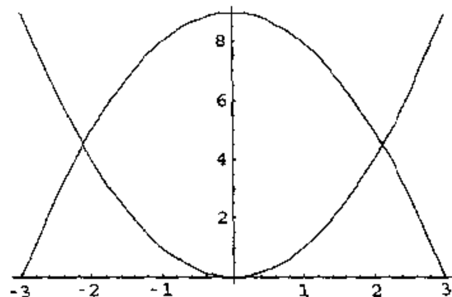
可以在同一个坐标系中画出两个函数的图形．

■　**Plot[{f[x], g[x]}, {x, xmin, xmax}]**在同一坐标系中绘制函数 $f(x)$ 与 $g(x)$ 在区间 xmin≤x≤xmax 上的图形．这条命令自然可以推广到绘制三个或更多函数图形的情形．

例 2　画出函数 $f(x)=x^2$ 与 $g(x)=9-x^2$ 在 －3 到 3 之间的图形．

```
Plot[{x^2, 9 - x^2}, {x, - 3, 3}];
```

←注意这里使用了分号．

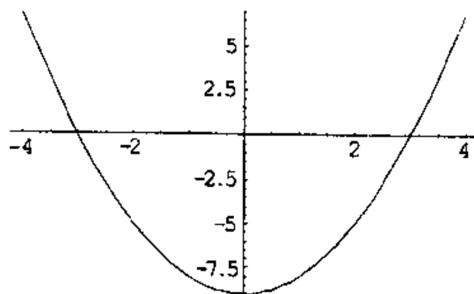


如果要同时显示几幅图形时,特别当图形的定义区间不同时,**Show** 命令是相当有用的.

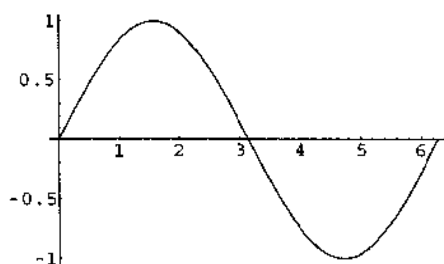
■ **Show[g1, g2, …]**在同一个坐标系中显示几幅图形.

例 3 假设希望作出函数 $y = x^2 - 9$ 在区间[−4, 4]上的图形和函数 $y = \sin x$ 在区间[0, 2π]上的图形,并且要把它们显示在同一坐标系中,那么进行下述操作.定义

```
g1 = Plot[x^2 - 9, {x, -4, 4}];
```

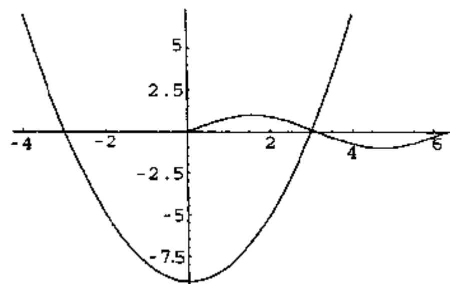



```
g2 = Plot[Sin[x], {x, 0, 2π}];
```



现在可以应用 **Show** 命令.注意为了同时显示两幅图形,坐标轴进行了调整.

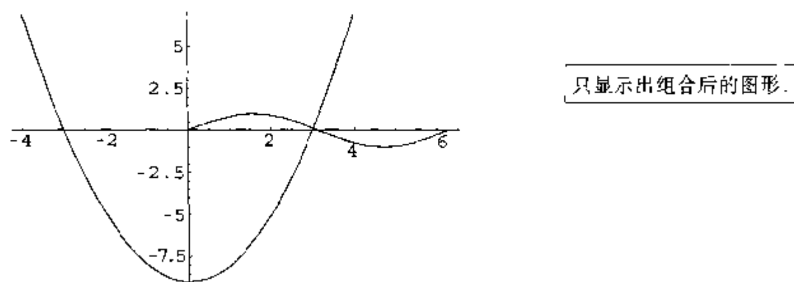
```
Show[g1, g2];
```



可见在前面这个例子构造 g1 与 g2 的过程中,每条曲线都首先单独显示在自己的坐标系中.为了禁止输出图形,可以用 **DisplayFunction** 选项.DisplayFunction 指定控制图形输出的函数.

- **DisplayFunction →Identity** 禁止输出图形.
- **DisplayFunction → $DisplayFunction** 恢复显示图形的能力.

例 4
```
g1 = Plot[x^2 - 9, {x, -4, 4}, DisplayFunction ->Identity];
g2 = Plot[Sin[x], {x, 0, 2π}, DisplayFunction ->Identity];
Show[g1, g2, DisplayFunction -> $DisplayFunction];
```



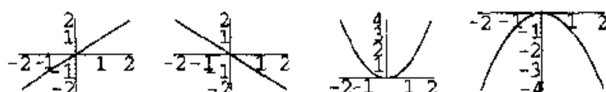
绘制多幅图形的另外一个实用命令是 **GraphicsArray**.

- **GraphicsArray[{g1, g2, …}]**绘制一行图形对象.
- **GraphicsArray[{{g11, g12, …}, {g21, g22, …}}]** 绘制图形对象的二维组.

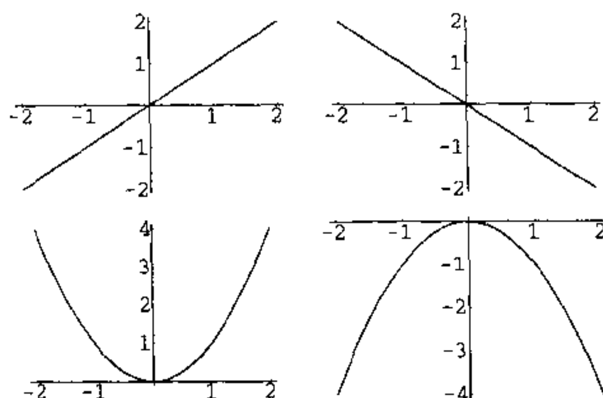
注意　与 Plot 不同,GraphicsArray 并不生成图形对象.因此应当用 Show 查看图形.

例 5

```
g1 = Plot[x, {x, -2, 2}, DisplayFunction→Identity];
g2 = Plot[-x, {x, -2, 2}, DisplayFunction→Identity];
g3 = Plot[x^2, {x, -2, 2}, DisplayFunction→Identity];
g4 = Plot[-x^2, {x, -2, 2}, DisplayFunction→Identity];
g = GraphicsArray[{g1, g2, g3, g4}];
Show[g, DisplayFunction→$DisplayFunction];
```



```
gg = GraphicsArray[{{g1, g2}, {g3, g4}}];
Show[gg, DisplayFunction→$DisplayFunction];
```

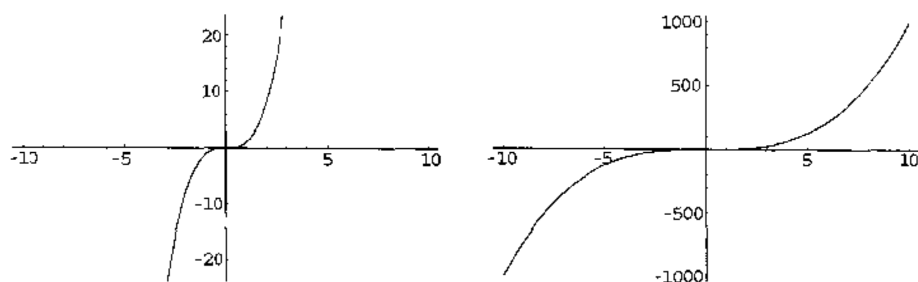


Plot 函数有许多选项,输入?? **Plot** 或者 **Options[Plot]**就可以看到它们到底有哪些.这些选项既可以单独使用,也可以组合在一起使用.在本节余下部分介绍其中相当常用的一些选项.

由于 Mathematica(显然)不可能绘制无穷多个点,因此它选取有限个等距点作为"采样点",并利用自适应算法构造出一条看起来光滑的曲线.它所使用的最大点数用 **PlotPoints** 表示,缺省值为 25.如果曲线相当摆动,那么就必须用更多的点.

- **PlotPoints →n** 指定在构造图形的过程中，所使用的采样点最大数目．
- PlotRange 选项指定在图形中应包含哪些点．
  - **PlotRange →Automatic** 为 Mathematica 的缺省值．如果由于某些点的纵坐标太大，而导致图形可能不太令人满意时（根据 Mathematica 标准），会自动从图形中删除它们．
  - **PlotRange→All** 强迫 Mathematica 画出所有的点．
  - **PlotRange→{ymin, ymax}** 只画出纵坐标介于 ymin 与 ymax 之间的那些点．
  - **PlotRange→{{xmin, xmax}, {ymin, ymax}}** 画出横坐标介于 xmin 与 xmax 之间，纵坐标介于 ymin 与 ymax 之间的点．

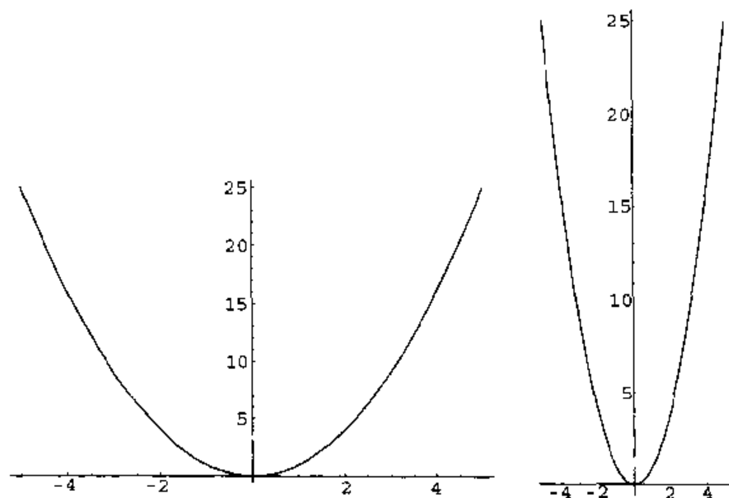
例 6　`Plot[x^3, {x, -10, 10}];Plot[x^3, {x, -10, 10}, PlotRange →All];`



在作图时，你会注意到横轴与纵轴通常采用不同长度．默认情况下纵轴长度与横轴长度比率为 1/GoldenRatio，其中 GoldenRatio = $(1+\sqrt{5})/2$．Mathematica 的设计人员认为这个比率是最好看的．可以通过选项 **AspectRatio** 改变这种方式，该选项定义图形的高宽比率．

- **AspectRatio →Automatic** 利用图形的实际坐标轴计算外观比率．
- AspectRatio →**比率**把纵轴与横轴的比率取为指定的比率值．

例 7　`Plot[x^2, {x, -5,5}];`
`Plot[x^2, {x, -5, 5}, AspectRatio →Automatic];`

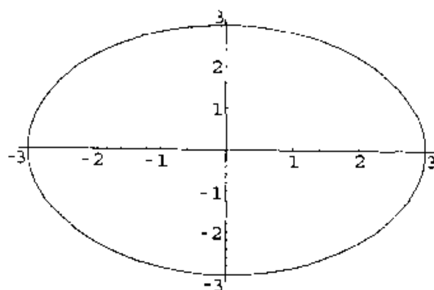


两个坐标轴的放缩倍数相同．

例 8　下面的命令应当生成一个圆心在原点的半径为 3 的圆．但是由于两个坐标轴的放缩倍

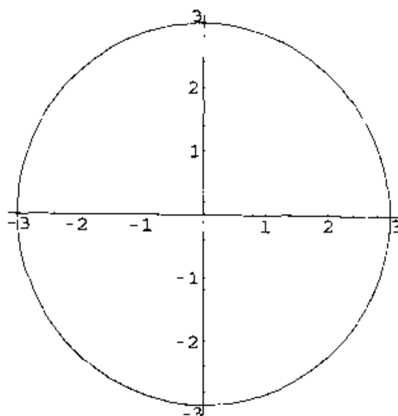
数不同,从而使得图形显示为椭圆.

```
Plot[{-Sqrt[9-x^2], Sqrt[9-x^2]}, {x, -3, 3}];
```



可以通过设置 AspectRatio→Automatic 使得上面的圆显示的更像一个圆.

```
Plot[{-Sqrt[9-x^2], Sqrt[9-x^2]}, {x, -3, 3}, AspectRatio→Automatic];
```

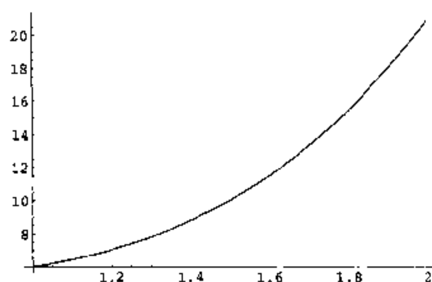


在绘制图形的时候,Mathematica 要确定原点的位置.如果 (0, 0) 在绘图区域的内部,那么坐标轴就在该点相交.否则,利用一个算法确定坐标轴应当在何处相交.在有些情形中,这可能会使函数的显示令人混淆(甚至误导).选项 AxesOrigin 可以控制坐标轴交点的位置.

AxesOrigin 是一个二维的图形选项,它确定坐标轴的交点.

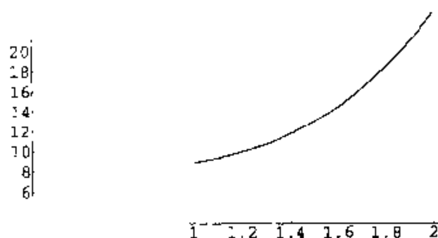
- **AxesOrigin→Automatic** 为默认值.如果点(0,0)位于绘图区域内,或者靠近这个区域,那么通常就取该点为坐标轴交点.
- **AxesOrigin→{x, y}** 强迫坐标轴交于(x, y)点.

例 9 `Plot[5 + x^4, {x, 1, 2}];`



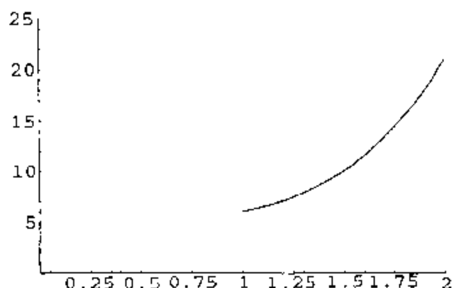
坐标轴交于(1,6)点。然而从 $x=1$ 到 $x=6$ 的图形被完整画出来.

```
Plot[5 + x^4, {x, 1, 2}, AxesOrigin→{0, 0}];
```



现在坐标轴交于 (0, 0) 点,但只有对应于指定区域的坐标轴部分被画出来.可以用 PlotRange 很容易地解决这个问题.

Plot[5 + x⁴, {x, 1, 2}, AxesOrigin→{0, 0}, PlotRange→{{0, 2}, {0, 25}}];



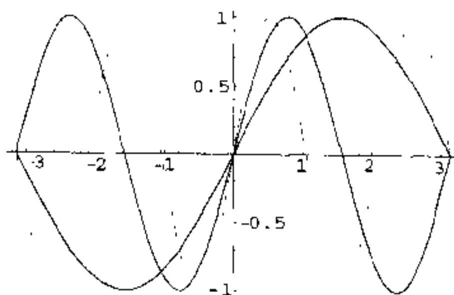
利用 **PlotStyle** 可以改变图形的外观.当在同一坐标系中同时绘制几幅图形的时候,这个选项是相当有用的.该选项的形式为

- **PlotStyle→样式** 如果只使用一个样式时,就采用这种方式.
- **PlotStyle→{样式 1, 样式 2, …}** 如果希望使用几种样式时,就采用这种方式.如果要修改的图形不只是一幅,那么样式循环作用到它们上面.

下面列出一些相当常用的绘图样式:

- **GrayLevel[x]**,其中 $0 \leqslant x \leqslant 1$,改变图像的颜色深浅.$x$ 的值越靠近 1,图像的颜色就越浅.

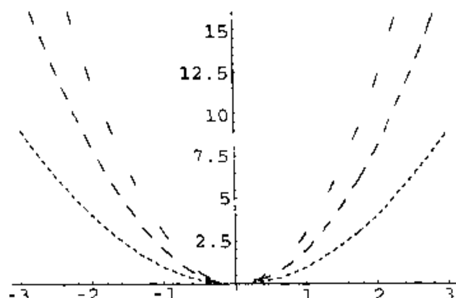
例 10 **Plot[{Sin[x], Sin[2x], Sin[3x]}, {x, -π, π},**
PlotStyle→{GrayLevel[0.0], GrayLevel[0.5], GrayLevel[0.8]}];



- **Dashing[{r1, r2, …, rm}]** 定义曲线的虚线方式,这里的参数值表示循环绘制线段与空白时的长度值.每个 r 值都表示占图形总宽的百分比.
- **AbsoluteDashing[a1, a2, …, am]** 定义曲线的虚线方式,这里的参数值表示循环绘制线段与空白时的绝对长度.每个绝对长度都以打印机点(近似为 1 英寸的 1/72)

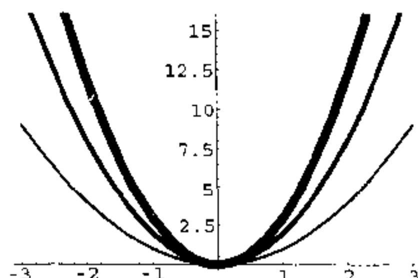
为单位.

例 11 Plot[{x^2, 2x^2, 3x^2}, {x, -3, 3}, PlotStyle→{Dashing[{.01}],
Dashing[{.03}], Dashing[{.03,.1}]}]



- **Thickness[r]**指定图形绘制的宽度 r.这里的宽度 r 为图形总宽度的百分比.二维图形的默认值是 0.004.
- **AbsoluteThickness[d]**指定图形绘制的绝对宽度 d.绝对长度以打印机点(近似为 1 英寸的 1/72)为单位.

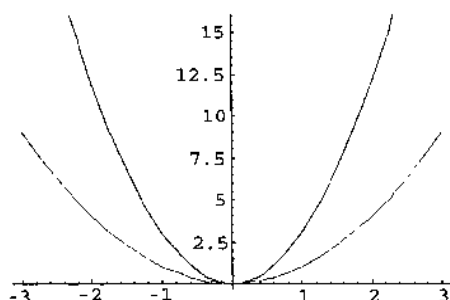
例 12 Plot[{x^2, 2x^2, 3x^2}, {x, -3, 3}, PlotStyle→{Thickness[.005],
Thickness[.01], Thickness[.02]}];



有几个图形选项可以使用彩色绘制图形.

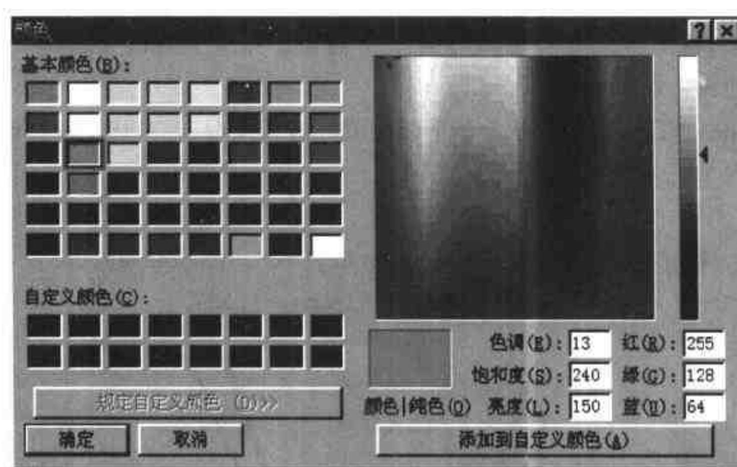
- **Hue[色调]** 指定一个颜色.当色调值从 0 变到 1 时,对应的颜色遍历红色、黄色、绿色、青色、蓝色、紫红色,并再次回到红色.
- **Hue[色调, 饱和度, 明亮度]** 用色调、饱和度以及明亮度一起表示颜色.饱和度与明亮度的值必须介于 0 到 1 之间.
- **RGBColor[红, 绿, 蓝]** 指定由红、绿、蓝的混合而得到的特定颜色.红、绿、蓝的值必须介于 0 到 1 之间.RGB[1, 0, 0]生成纯红色结果;相应地 RGBColor[0, 1, 0]生成纯绿色,RGBColor[0, 0, 1]生成纯蓝色.
- **CMYKColor[青, 紫红, 黄, 黑]** 指定由青、紫红、黄、黑的混合而得到的特定颜色.青、紫红、黄、黑的值必须介于 0 到 1 之间.CMYKColor 是用于打印到纸张的颜色.

例 13 Plot [{x^2, 2x^2, 3x^2}, {x, -3, 3}, PlotStyle→{RGBColor[1, 0, 0],
RGBColor[0, 1, 0], RGBColor[0, 0, 1]}];

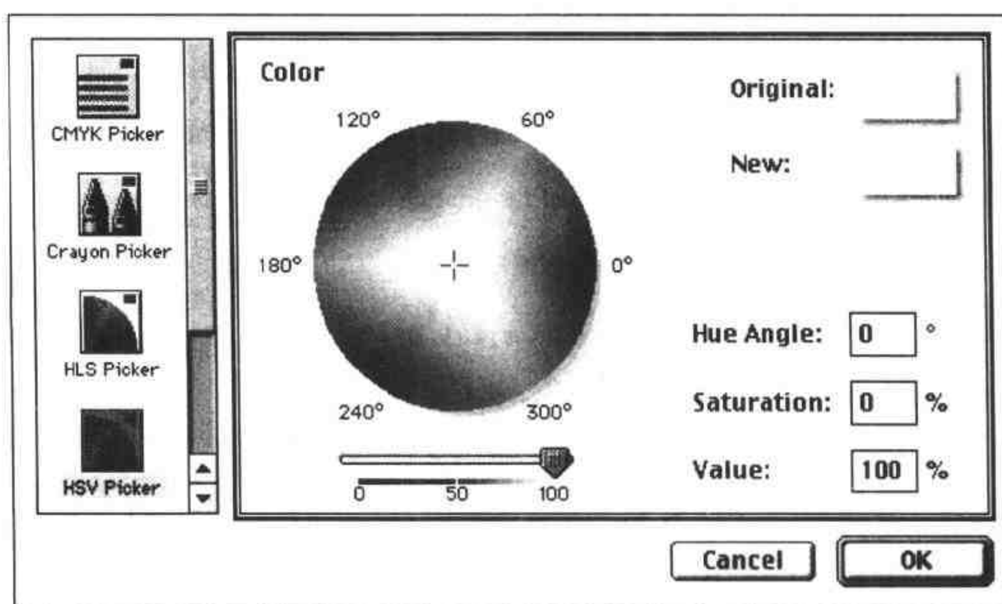


这里 $x^2, 2x^2, 3x^2$ 的图像分别显示为红色、绿色和蓝色.

在 Mathematica 中可以非常容易地计算各种颜色的 RGB 值. 只要进入 **Input ⇒Color Selector**, 并点击所希望的颜色, 就可以达到目的. 选定颜色的精确 RGB 值组合就会插入到 Mathematica 笔记本的插入点所处的位置.



微机上的颜色选择窗口



Macintosh 机上的颜色选择窗口

软件包 **Graphics`Colors`** 中包含一组预先定义的颜色.在上载了该软件包后输入 **AllColors** 就可以查看到所有颜色的名称列表.一旦选定了一种颜色,就可以在 **RGBColor** 指定颜色的地方直接使用该颜色的名称.如果要查看它对应的 RGB 公式,只要输入颜色的名称就可以了.

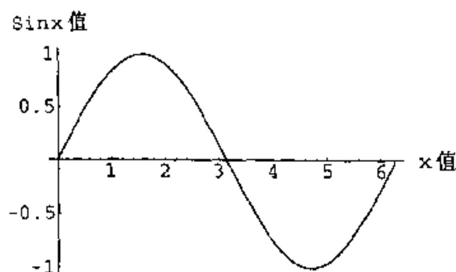
例 14
```
<<Graphics`Colors`
AliceBlue
RGBColor[0.941206, 0.972503, 1.]
```

用两种选项可以用来给图形加标签.其中 **PlotLabel** 指定图形的整体标签,而 **AxesLabel** 可以给一个或两个坐标轴加上各自的适当描述性标签.

- **PlotLabel →描述** 给图形加上标题性标签.
- **AxesLabel →标签** 只为 y 轴指定标签.
- **AxesLabel →{x 轴标签, y 轴标签}** 指定 x 轴与 y 轴的标签.
- **AxesLabel →None** 指定不给坐标轴加标签.这是 Mathematica 的默认值.

上面的描述、标签、x 轴标签以及 y 轴标签可以是任意的 Mathematica 符号或字符串(即包围在双引号内的文本).如果同时使用 PlotLabel 和 AxesLabel,就必须仔细一些,以防标签互相重叠.

例 15
```
Plot[Sin[x], {x, 0, 2π},
    AxesLabel->{"Values of x", "Values of sin x"}];
```



PlotLegend 是一个给图形加标签的相当有用的选项.然而,它是包含在软件包 Graphics`Legend` 中的,因此如果要使用该选项,必须先上载这个软件包.

- **PlotLegend[{文本 1, 文本 2, …}]** 给 PlotStyle 中指定的每个描述附加上文本 1,文本 2, ….如果 PlotStyle 中的描述数目多于文本描述,那么文本描述按循环方式重复使用.

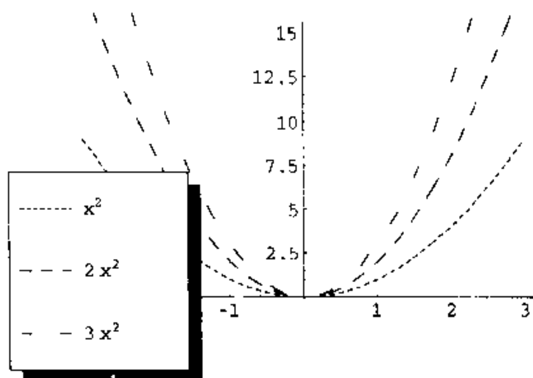
例 16
```
<<Graphics`Legend`
Plot[{x^2, 2x^2, 3x^2}, {x, -3, 3}, PlotStyle→{Dashing[{.01}],
Dashing[{.03}], Dashing[{.03, .08}]}, PlotLegend→{x^2, 2x^2, 3x^2}]
```

如果必要的话,可以用矩形框包围起图形.另外,也可以不显示一个或两个坐标轴.

Frame 指定是否应在图形四周加上方框.

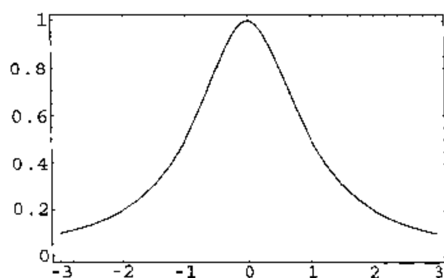
- **Frame →True** 指定在图形的四周加上一个矩形方框.
- **Frame →False** 指定不在图形的四周加上一个矩形方框(默认值).

Axes 确定是否应画出坐标轴.



- **Axes →True** 指定要画出两个坐标轴(默认值).
- **Axes →False** 不画出坐标轴.
- **Axes →{False, True}** 画出 y 轴但不画出 x 轴.
- **Axes →{True, False}** 画出 x 轴但不画出 y 轴.

例 17　**Plot[$\frac{1}{x^2+1}$, {x, - 3, 3}, Frame →True, Axes →False];**

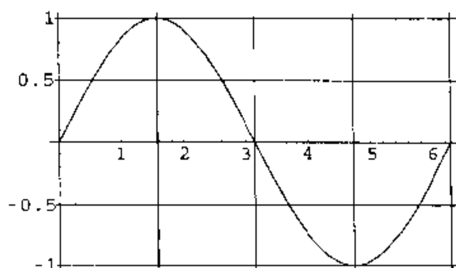


GridLines 指定在图形中绘制矩形网格.

- **GridLines →None** 不绘制网格线(默认值).
- **GridLines →Automatic** 绘制网格线,位置由 Mathematica 确定.
- **GridLines →{*x* 列表, *y* 列表}** 要求网格线出现在指定的位置上.这里的列表分别为一组数或者为 Automatic,对于后者,表示由 Mathematica 确定网格线的位置.

例 18　在绘制三角函数的图形时,把竖直格线放在 π/2 的倍数处是非常方便的.

Plot[Sin[x], {x, 0, 2π}, GridLines →{{0, π/2, π, 3π/2, 2π}, Automatic}];



在坐标轴上的刻度线以及标记可以用选项 **Ticks** 控制. 当 Frame→True 被设置时, **FrameTicks** 提供在矩形框边界上类似的选项控制.

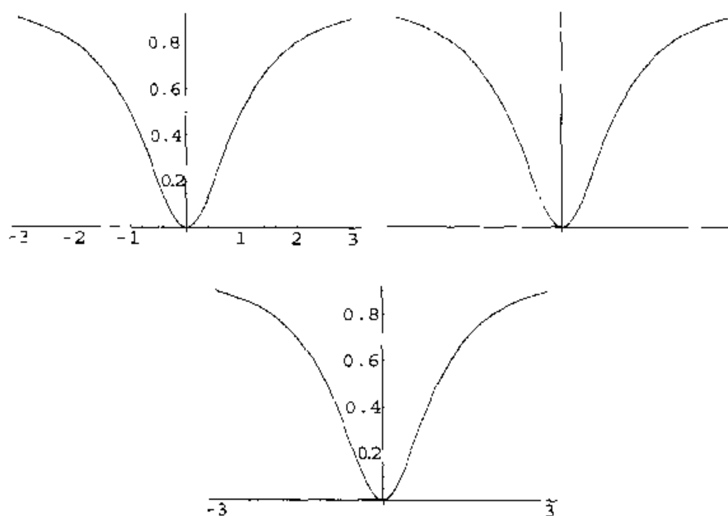
- **Ticks →None** 指定不画出刻度线, 也不显示数字标记.
- **Ticks →Automatic** 指定画出刻度线. 这是默认值.
- **Ticks →{ x 列表, y 列表}**要求在指定位置处画出刻度线, 这里的列表分别为一组数或者为 Automatic.

例 19 下面是绘制 $y=\frac{x^2}{x^2+1}$ 图形的三种方法.

Plot[$\frac{x^2}{x^2+1}$, {x, -3, 3}];

Plot[$\frac{x^2}{x^2+1}$, {x, -3, 3}, Ticks →None];

Plot[$\frac{x^2}{x^2+1}$, {x, -3, 3}, Ticks →{{-3, 3}, Automatic}];



FilledPlot 命令绘制加阴影的图形, 这条命令在软件包 **Graphics\`FilledPlot\`** 中, 因此必须先上载这个软件包才能使用该命令.

- **FilledPlot[f[x], {x, xmin, xmax}]** 画出 f[x]的图形, 并给由函数与 x 轴围成的区域加上阴影.
- **FilledPlot[{f1[x], f2[x], …}, {x, xmin, xmax}]** 画出 f1[x], f2[x], …的图形, 并给由相邻图形围成的区域加上不同颜色的阴影.

例 20 **≪Graphics\`FilledPlot\`**

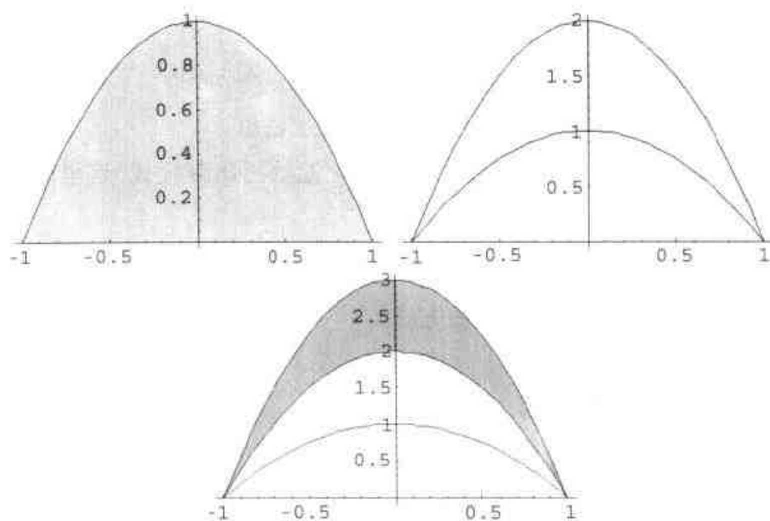
FilledPlot[1 - x^2, {x, -1, 1}];

FilledPlot[{1 - x^2, 2 - 2x^2}, {x, -1, 1}];

FilledPlot[{1 - x^2, 2 - 2x^2, 3 - 3x^2}, {x, -1, 1}];

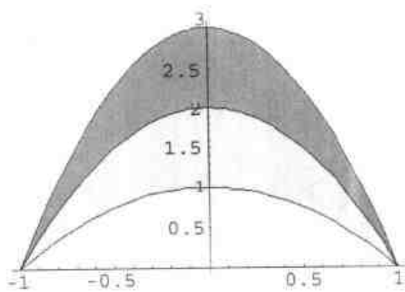
任何标准的图形选项都可以用在 FilledPlot 中. 另外,

- **Fills →{填充样式 1, 填充样式 2, …}** 指定相邻曲线由填充样式 1, 填充样式 2, …填充.



例 21

```
<<Graphics`FilledPlot`
FilledPlot[{1 - x^2, 2 - 2x^2, 3 - 3x^2}, {x, -1, 1},
  Fills -> {GrayLevel[.8], GrayLevel[.6], GrayLevel[.3]}];
```

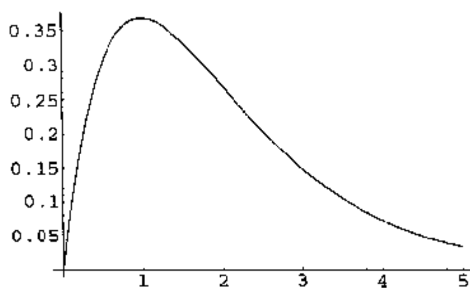


习题解答

4.1 画出函数 $y = xe^{-x}$ 在 $x=0$ 与 $x = 5$ 之间的图形.

解 ☞

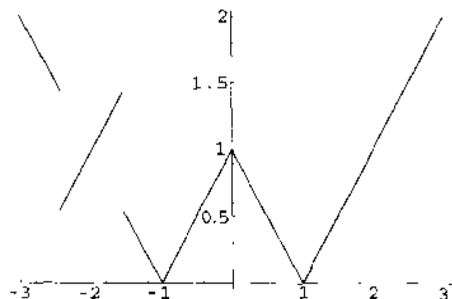
```
Plot[x Exp[-x], {x, 0, 5}];
```



4.2 画出函数 $f(x) = |1 - |x||$ 在区间 $[-3, 3]$ 上的图形.

解 ☞

```
Plot[Abs[1 - Abs[x]], {x, -3, 3}];
```



4.3 在概率统计中标准的正态曲线就是函数

$$f(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$$

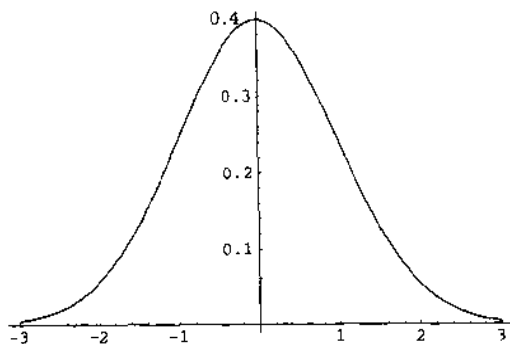
的图形. 画出这个函数在 $-3 \leqslant x \leqslant 3$ 上的图形.

解 ☞

```
f[x_] = 1/(Sqrt[2 * Pi]) * Exp[(-1/2)x^2];
```

或者 $\mathbf{f[x_]} = \frac{1}{\sqrt{2\pi}}\mathbf{Exp}[-\frac{1}{2}\mathbf{x}^2]$;

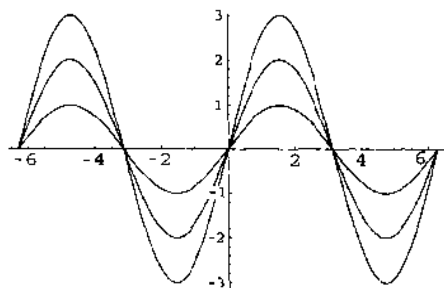
```
Plot[f[x], {x, -3, 3}];
```



4.4 在同一个坐标系中画出函数 $y = \sin x$, $y = 2\sin x$ 与 $y = 3\sin x$ 从 -2π 到 2π 的图形.

解 ☞

```
Plot[{Sin[x], 2Sin[x], 3Sin[x]}, {x, -2π, 2π}];
```

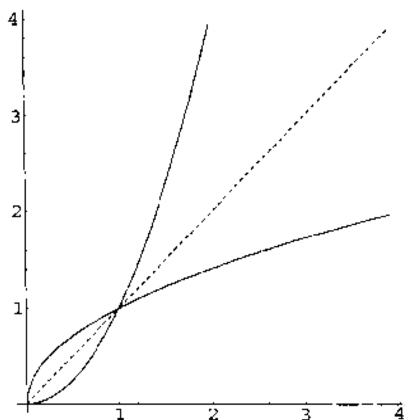


4.5 反函数的图形与原函数的图形关于直线 $y = x$ 是对称的. 用实线画出互为反函数的函数

$f(x)=x^2$ 与 $f^{-1}(x)=\sqrt{x}$, $0\leqslant x\leqslant 1$ 的图形,而用虚线画出 $y=x$ 的图形,以观察对称性.

解 ☞

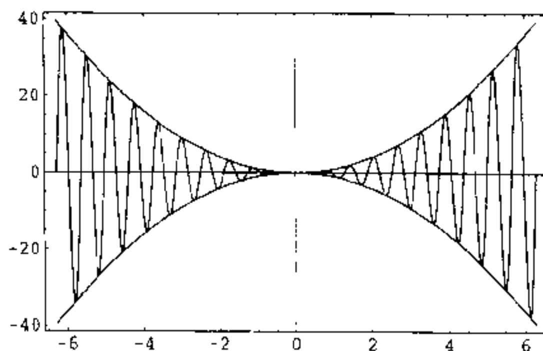
```
g1 = Plot[x², {x, 0, 2},DisplayFunction →Identity];
g2 = Plot[√x, {x, 0, 4},DisplayFunction →Identity];
g3 = Plot[x, {x, 0, 4},PlotStyle →Dashing[{0.01}], DisplayFunction →Identi-
    ty];
Show [g1, g2, g3, AspectRatio → Automatic, DispalyFunction → $Display-
Function];
```



4.6 在同一个坐标系中画出 $y=x^2$, $y=-x^2$, $y=x^2\sin 10x$, 在区间 $-2\pi\leqslant x\leqslant\pi$ 上的图形,并用矩形框包围起来.

解 ☞

```
Plot[{x², - x², x²Sin[10x]}, {x, - 2π,2π}, Frame →True];
```



4.7 在逼近论和数值分析中经常用到切比雪夫多项式族.在 Mathematica 中这些多项式的表示为 **ChebyshevT[n, x]**.在同一个坐标系中,利用不同的方法区别开各条曲线,画出次数分别为 2, 3, 4 的切比雪夫多项式的图形,并加上标签.

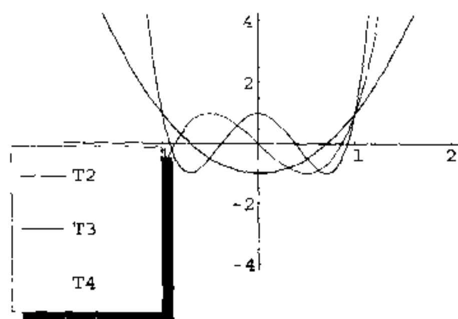
解 ☞

```
<<Graphics`Legend`
Plot [{ChebyshevT[2, x], ChebyshevT[3, x], ChebyshevT[4, x]},
    {x, - 2, 2}, PlotStyle →{GrayLevel[0], GrayLevel[.4],
    GrayLevel[.7]}, PlotLegend →{T2, T3, T4}];
```

另外的解答

```
<<Graphics`Legend`
Plot [{ChebyshevT[2, x], ChebyshevT[3, x], ChebyshevT[4, x]},
    {x, -2, 2}, PlotStyle→{RGBColor[1, 0, 0], RGBColor[0, 1, 0],
    RGBColor[0, 0, 1]}, PlotLegend→{T2, T3, T4}];
```

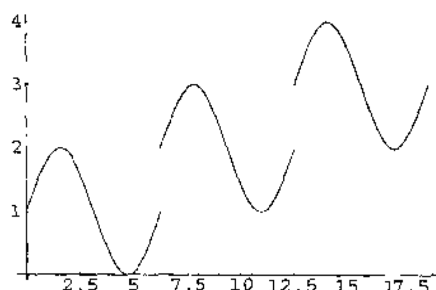
彩色图形不显示



4.8　在同一个坐标系中画出 $y=1+\sin x, 0\leqslant x\leqslant 2\pi$, $y=2+\sin x, 2\pi\leqslant x\leqslant 4\pi$ 与 $y=3+\sin x, 4\pi\leqslant x\leqslant 6\pi$ 的图形.

解

```
g1 = Plot[1 + Sin[x], {x, 0, 2π}, DisplayFunction→Identity];
g2 = Plot[2 + Sin[x], {x, 2π, 4π}, DisplayFunction→Identity];
g3 = Plot[3 + Sin[x], {x, 4π, 6π}, DisplayFunction→Identity];
Show[g1, g2, g3, DisplayFunction→ $DisplayFunction];
```



4.2　其它的绘图命令

标准的几何形状可以用 Graphics 命令进行构造,并用 Show 命令查看.

■ Graphics[基本单元] 利用图形基本单元构造一幅图形.

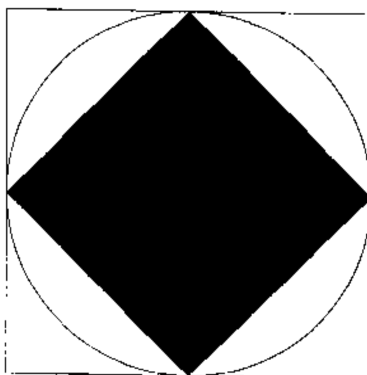
下面列出的是在 Mathematica 中比较常用的几个图形基本单元.

- Circle[{x, y}, r] 创建一个圆,圆心在(x, y)处,半径为 r.
- Disk[{x, y}, r] 创建一个圆盘(即被填充的圆),圆心在(x, y),半径为 r.
- Point[{x, y}]在坐标(x, y)处画一个点.
- Line[{{x1, y1}, {x2, y2}, …}] 绘制连接(x1, y1),(x2, y2),…等点的线段.
- Rectangle[{x1, y1}, {x2, y2}] 创建一个被填充的矩形,其中(x1, y1)与(x2, y2)为所构造矩形的对角线两个端点.

- **Polygon**[{{**x1, y1**}, {**x2, y2**}, …}] 构造一个被填充的多边形,(x1, y1),(x2, y2),…等为顶点.
- **Text**[**文本字符串**, {**x, y**}] 在(x, y)位置上显示文本字符串.

当用 Show 查看图形对象时,Axes 的默认值为 Axes→**False**.如果必要的话,可以在选项中包含 Axes→True 以显示坐标轴.

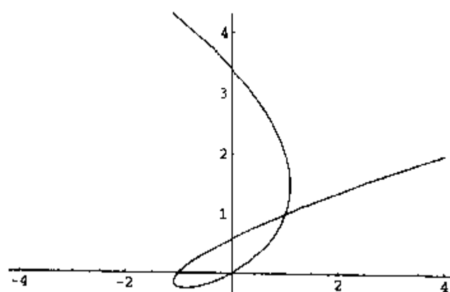
例 22
```
g1 = Graphics[Circle[{0, 0}, 1]];
g2 = Graphics[Line[{{-1, -1}, {-1, 1}, {1, 1}, {1, -1}, {-1, -1}}]];
g3 = Graphics[Polygon[{{-1, 0}, {0, 1}, {1, 0}, {0, -1}}]];
Show[g1, g2, g3, AspectRatio →Automatic];
```



有时候曲线的定义采用的是参数形式,即点的 x 与 y 坐标定义为关于第三个变量的两个独立函数.参数曲线通常比较复杂,可以用 **ParametricPlot** 查看其形状.

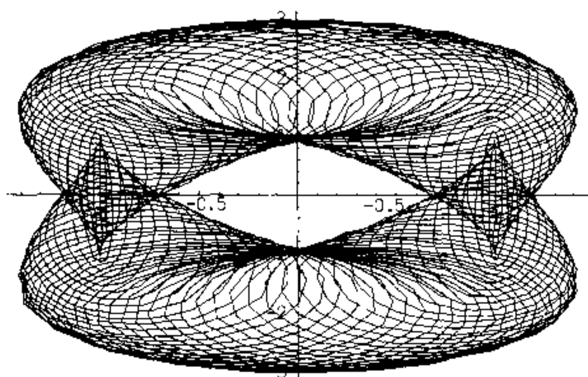
- **ParametricPlot**[{**x**[**t**], **y**[**t**]}, {**t, tmin, tmax**}] 在区间 tmin≤t≤tmax 上画出参数方程 $x = x(t), y = y(t)$ 的图形.
- **ParametricPlot**[{{**x**[**t**], **y**[**t**]}, {**x2**[**t**], **y2**[**t**]}, …}, {**t, tmin, tmax**}] 在区间 tmin≤t≤tmax 上画出几条参数方程的图形.

例 23 **ParametricPlot**[{t^3 - 2t, t^2 - t}, {t, -2, 2}];



例 24
```
x[t_] = Cos[t] - Cos[100t] Sin[t];
y[t_] = 2 Sin[t] - Sin[100t];
ParametricPlot[{x[t], y[t]}, {t, 0, 2π}, PlotPoints →50];
```

隐式函数的图形可以用 **ImplicitPlot** 命令绘制.由于这条命令包含在软件包 **Graphics`ImplicitPlot`** 中,因此在使用前必须上载这个软件包.



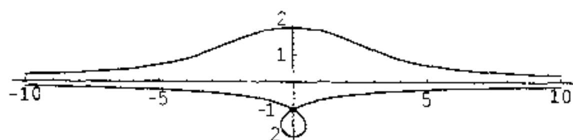
- **ImplicitPlot[方程, {x, xmin, xmax}]** 画出当 x 的值从 xmin 变到 xmax 时方程的图形. 对于 x 的每个值, 计算出 y 的值, 并画出所得到的几个点(x, y).
- **ImplicitPlot[方程, {x, xmin, xmax}, {y, ymin, ymax}]** 在绘制方程图形时, 认为它是三维空间中的函数, 通过用 z 等于零的平面与其三维图形相交得到所定义的二维曲线. 这种方法速度快, 但可能得到比较粗糙的图形, 特别是在图形的奇异点或交点处表现得更突出.
- **ImplicitPlot[方程, {x, xmin, x1, x2, …, xmax}]** 画出方程除 x1, x2, …等点外所定义的图形, 因为这些点有可能导致麻烦.

上面所指的**方程**必须是 lhs == rhs 的形式. 注意中间用的双等号.

由于需要大量的计算, 因此 ImplicitPlot 为了画出图形, 会花费一些时间. 命令的第二种形式(即轮廓方法)速度要快些.

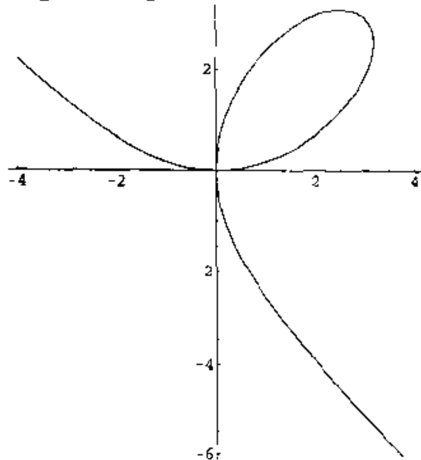
例 25 画出方程 $x^2y^2 = (y+1)^2(4-y^2)$ 在 $-10 \leqslant x \leqslant 10$ 时定义的曲线. (这条曲线称为 Nicomedes 蚌线.)

```
<<Graphics`ImplicitPlot`
ImplicitPlot[x^2 y^2 == (y+1)^2(4-y^2), {x, -10, 10},
                                PlotRange -> All];
```



例 26 画出方程 $x^3 + y^3 = 6xy$ 在 $-4 \leqslant x \leqslant 4$ 时的图形. (这条曲线称为笛卡儿叶形线.)

```
<<Graphics`ImplicitPlot`
ImplicitPlot[x^3 + y^3 == 6x y, {x, -4, 4}];
```



在下面这个例子中，我们必须使用轮廓方法，因为 ImplicitPlot 不能求解包含超越函数的方程．为此，我们需要除指定 x 的区间外，还要指定 y 的区间．

例 27　画出 $\cos(x-y)-y\sin x$，$-2\leqslant x\leqslant 2$ 的图形．

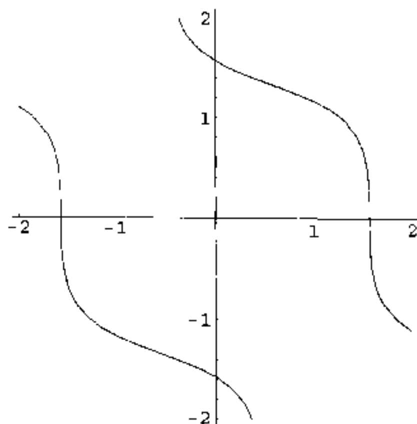
```
<<Graphics`ImplicitPlot`
ImplicitPlot[Cos[x-y] == y Sin[x], {x, -2,2},AxesOrigin->{0, 0}];
Solve::tdep: The equations appear to involve transceddental
  functions of the variables in an essentially non-algebraic way.
```

（给定的方程中包含超越函数，其中的变量以非代数方式出现．）

```
ImplicitPlot::epfail:
    Cos[x-y] == ySin[x] could not be solved for points to plot.
```

（不能从方程 Cos[x - y] == y Sin[x]中解出画图的点．）

```
ImplicitPlot[Cos[x-y] == y Sin[x], {x, -2, 2}, {y, -2, 2},
                    AxesOrigin->{0, 0}];
```



对于用极坐标定义的曲线，可以用软件包 Graphics`Graphics` 中的 **PolarPlot** 命令绘制其图形．

- **PolarPlot[f[θ], {θ, θmin, θmax}]** 画出极坐标方程 $r=f(\theta)$ 当 θ 从 θmin 变到 θmax 时的图形．

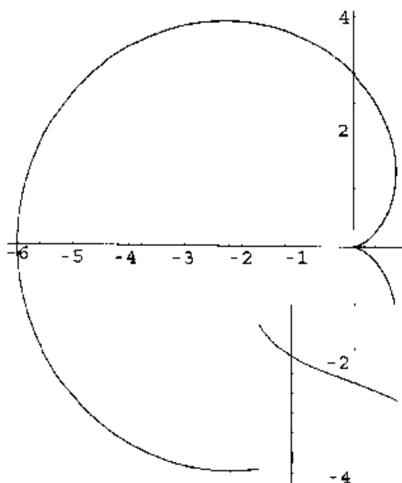
- **PolarPlot[{f1[θ], f2[θ], …}, {θ, θmin, θmax}]** 在同一个坐标系中画出几个极坐标方程的图形．

注意　对于 PolarPlot，默认外观比率为 AspectRatio->Automatic.

例 28

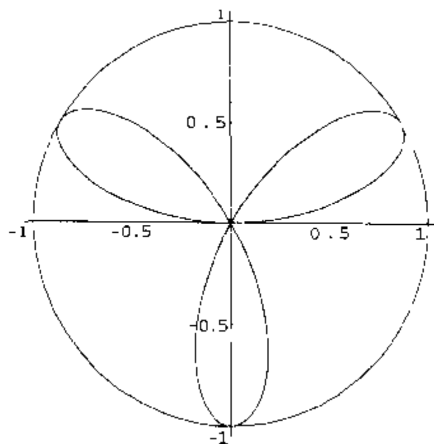
```
<<Graphics`Graphics`
PolarPlot[3 (1 - Cos[θ]), {θ,
0, 2π}];
```

（这条曲线称为心脏线．）



例 29　在单位圆 $r=1$ 内画出三叶玫瑰线 $r=\sin 3\theta$ 的图形.

```
<<Graphics`Graphics`
PolarPlot[{1, Sin[3θ]}, {θ, 0, 2θ}];
```

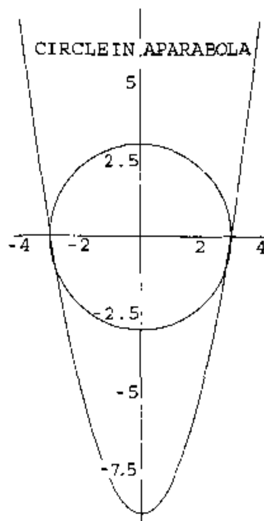


习 题 解 答

4.9　画出抛物线 $y=x^2-9$ 以及半径为 3,圆心在原点的圆的图形.

解 ☞

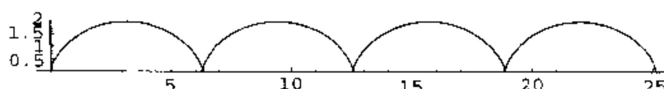
```
g1 = Plot[x^2 - 9, {x, -4, 4}, DisplayFunction →Identity];
g2 = Graphics[Circle[{0, 0}, 3]];
g3 = Graphics[Text["CIRCLE IN A PARABOLA", {0, 6}]];
Show [g1, g2, g3, AspectRatio →Automatic,
    DisplayFunction → $DisplayFunction];
```



4.10 当圆沿直线滚动时，圆上一点形成的轨迹称为旋轮线或摆线，其参数方程为 $x = r(\theta - \sin\theta)$, $y = r(1 - \cos\theta)$，其中 r 表示圆的半径. 画出当圆的半径为 1 时所形成的旋轮线的四拱.

解 ☞

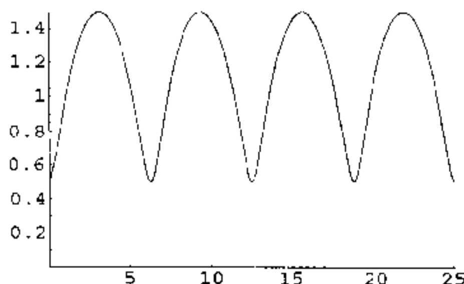
```
ParametricPlot[{θ - Sin[θ], 1 - Cos[θ]}, {θ, 0, 8π},
                         AspectRatio →Automatic];
```



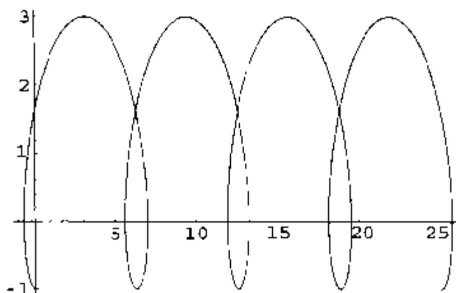
4.11 设 P 为距离半径为 r 的圆的圆心距离为 a 的点.（不妨想像该点为自行车轮辐条上的一点.）当圆沿直线滚动时，P 点形成的轨迹称为次摆线. 它的参数方程为 $x = r\theta - a\sin\theta$、$y = r - a\cos\theta$. 画出当 $r = 1$, $a = 1/2$ 时次摆线的四个周期. 如果 $r = 1$, $a = 2$，即点在圆外面时，次摆线的形状又是怎样的呢？

解 ☞

```
r = 1;
a = 1/2;
ParametricPlot[{rθ - a Sin[θ],r - a Cos[θ]}, {θ,0,8π},
    AxesOrigin →{0, 0}, PlotRange →{{0, 8π}, {0, 1.5}}];
```



```
a = 2;
ParametricPlot[{rθ - a Sin[θ],r - a Cos[θ]}, {θ, 0, 8π},
    AxesOrigin →{0, 0}];
```



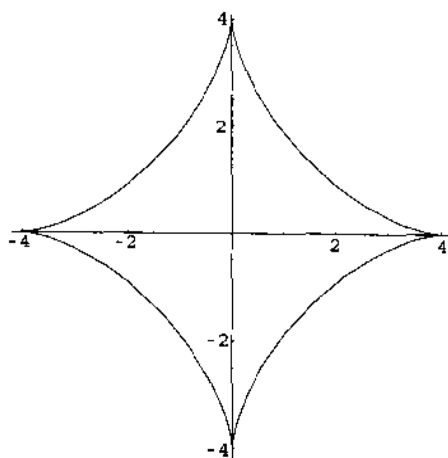
4.12 半径为 b 的圆在半径为 a 的大圆内边上滚动. 初始位置为 $(a, 0)$ 的固定点所形成的轨迹称为圆内旋轮线或内摆线，其方程为

$$\begin{cases} x = (a-b)\cos\theta + b\cos\left(\dfrac{a-b}{b}\theta\right) \\ y = (a-b)\sin\theta - b\sin\left(\dfrac{a-b}{b}\theta\right). \end{cases}$$

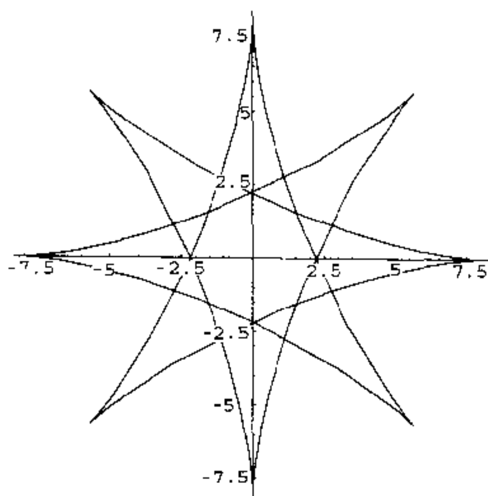
画出当 $a=1, b=4$ 以及 $a=8, b=5$ 时的内摆线.

解 ☞

```
x[θ_]:=(a-b) Cos[θ] + b Cos[(a - b)θ/b]
y[θ_]:=(a-b) Sin[θ] - b Sin[(a - b)θ/b]
a=4;
b=1;
ParametricPlot[{x[θ], y[θ]}, {θ, 0, 2π}, AspectRatio→Automatic];
```



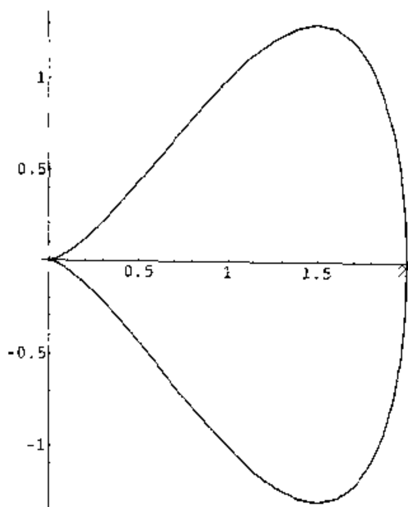

```
a=8;
b=5;
ParametricPlot[{x[θ],y[θ]}, {θ, 0, 10π},AspectRatio→Automatic];
```



4.13　画出由方程 $y^2 = x^3(2-x), 0 \leqslant x \leqslant 2$ 定义的曲线的图形.

解 ☞

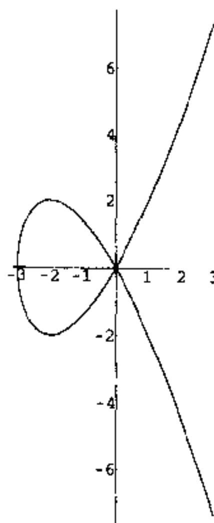
```
<<Graphics`ImplicitPlot`
ImplicitPlot[y^2 == x^3(2 - x), {x, 0, 2}];
```



4.14 画出 Tschirnhauser 三次曲线 $y^2=x^3+3x^2$, $-3\leqslant x\leqslant 3$ 的图形.

解 ☞

```
<<Graphics`ImplicitPlot`
ImplicitPlot[y^2 == x^3 + 3x^2, {x, -3, 3}];
```

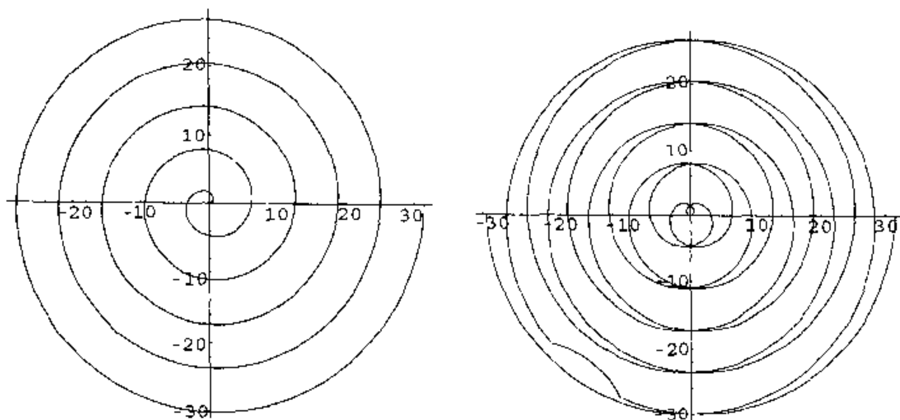


4.15 极坐标方程 $r=\theta$ 定义的曲线称为阿基米德(Archimedes)螺线.画出曲线当 $0\leqslant\theta\leqslant 10\pi$ 以及 $-10\pi\leqslant\theta\leqslant 10\pi$ 时的图形.

解 ☞

```
<<Graphics`Graphics`
PolarPlot[θ, {θ, 0, 10π}];
```

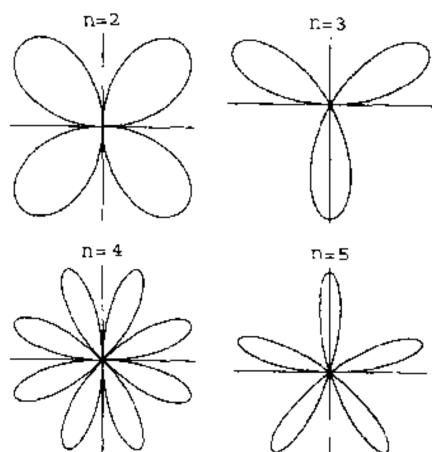
```
PolarPlot[θ, {θ, -10π, 10π}];
```



4.16 方程 $r=\sin n\theta$ 定义的极坐标曲线称为玫瑰线，其中 n 为正整数. 研究这族曲线的形状，并猜测循环的数目与 n 的关系.

解 ☞

```
<<Graphics`Graphics`
g1 = PolarPlot[Sin[2θ], {θ, 0, 2π}, Ticks→False,
        PlotLabel→"n = 2", DisplayFunction→Identity];
g2 = PolarPlot[Sin[3θ], {θ, 0, 2π}, Ticks→False,
        PlotLabel→"n = 3", DisplayFunction→Identity];
g3 = PolarPlot[Sin[4θ], {θ, 0, 2π}, Ticks→False,
        PlotLabel→"n = 4", DisplayFunction→Identity];
g4 = PolarPlot[Sin[5θ], {θ, 0, 2π}, Ticks→False,
        PlotLabel→"n = 5", DisplayFunction→Identity];
g = GraphicsArray[{{g1, g2}, {g3, g4}}];
Show[g, DisplayFunction → $DisplayFunction];
```



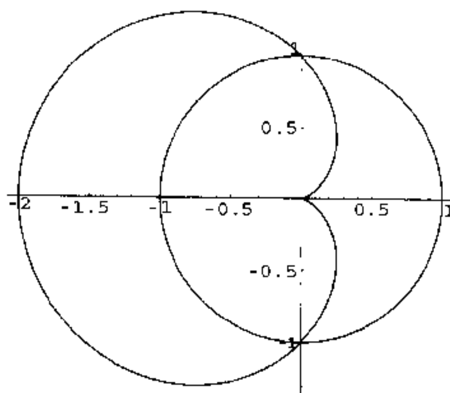
结论 如果 n 为奇数，玫瑰线有 n 叶；如果 n 为偶数，玫瑰线会有 $2n$ 叶.

4.17 在同一坐标系中画出心脏线 $r=1-\cos\theta$ 以及半径 $r=1$ 的圆.

解 ☞

```
<<Graphics`Graphics`
```

```
PolarPlot[{1 - Cos[θ], 1}, {θ, 0, 2π}];
```



4.3 特殊的二维绘图函数

所谓离散函数，就是定义在离散点集上的函数，可以用特殊的绘图函数 **ListPlot** 查看其形状.

■ **ListPlot[{y1, y2, …}]** 绘制点，其 y 坐标为 y1, y2, …，而 x 坐标则是正整数 1, 2, ….

■ **ListPlot[{{x1, y1}, {x2, y2}, …}]** 绘制点(x1, y1), (x2, y2), ….

在这个函数中可以使用标准的图形选项，从而命令的形式为

■ **ListPlot[{y1, y2, …}, 选项]**

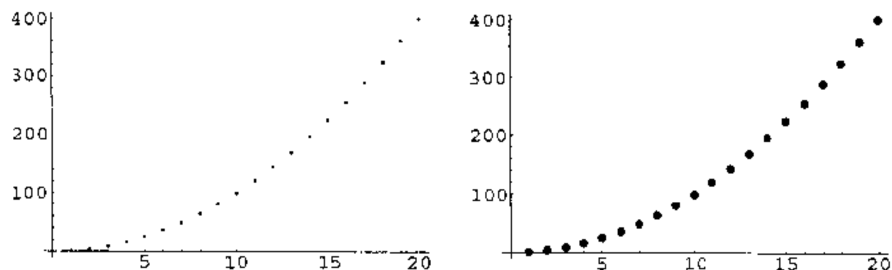
■ **ListPlot[{{x1, y1}, {x2, y2}, …}, 选项]**

在 **ListPlot** 中最常用的图形选项有：

- **PlotStyle→PointSize[d]**. 这里 d 指定点的直径，它表示相应于图形宽度的百分比. 默认值为 .008.
- **PlotStyle→AbsolutePointSize[d]**. 这里 d 表示点的绝对直径，它以打印机点为单位，一点约为一英寸的 1/72.
- **PlotJoined→True** 把点连接起来.

例 30 下面的绘图命令画出从 1 到 20 的连续正整数的平方.

```
squares = Table[k^2, {k, 1, 20}];
ListPlot[squares];
ListPlot[squares, PlotStyle→PointSize[.02]];
```



另外，**FilledListPlot** 将填充图形下面的区域，从而形成描述数据的填充多边形. 该命令的格式与 **ListPlot** 的相同. FilledListPlot 包含在 Graphics\`FilledPlot\` 中，因此在使用该命令前必须上载这个软件包.

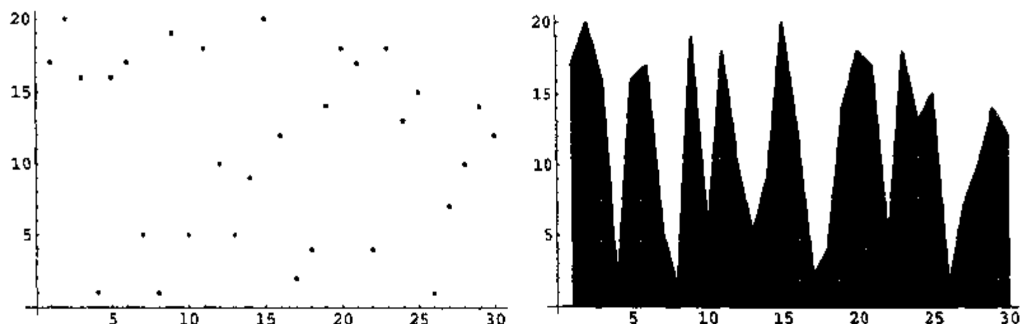
■ **FilledListPlot[{y1, y2, …}]** 绘制一个填充的点图，其 y 坐标为 y1, y2, …，而 x

坐标则是正整数 1, 2, ….

■ **FilledListPlot[{{x1, y1}, {x2, y2}, …}]** 绘制填充的点图,各点的坐标为(x1, y1), (x2, y2), ….

例 31
```
<<Graphics`FilledPlot`
randomintegers = Table[Random[Integer, {1, 20}], {k, 1, 30}];
ListPlot[randomintegers];
FilledListPlot[randomintegers];
```

> **Table** 命令生成由 30 个随机数构成的列表,每个随机数介于 1 与 20 之间.



可以用 **MultipleListPlot** 同时绘制几个列表的图形,该命令的语法与 **ListPlot** 类似,只是这里有可能指定几个列表.MultipleListPlot 包含在软件包 **Graphics`MultipleListPlot`** 中,因此在使用前须上载这个软件包.

■ **MultipleListPlot[列表 1, 列表 2, …]**

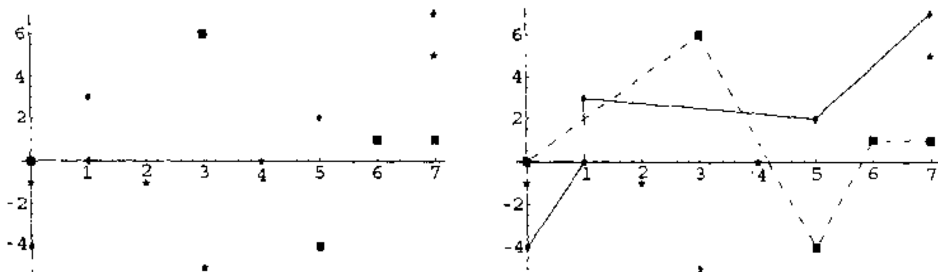
■ **MultipleListPlot[列表 1, 列表 2, …, 选项]**

这里的列表形式为{{x1, y1}, {x2, y2}, …}或者{y1, y2, …},其中,x 坐标取正整数 1, 2, ….

每个列表的图形用不同的符号表示.**PlotJoined** 选项确定是否把点连接起来.

• **PlotJoined →True** 用不同的线段把点连接起来.

例 32
```
<<Graphics`MultipleListPlot`
lst1 = {{0, -4}, {1, 0}, {1, 3}, {5, 2}, {7, 7}};
lst2 = {{0, -1}, {2, -1}, {3, -5}, {4, 0}, {7, 5}};
lst3 = {{0, 0}, {3, 6}, {5, -4}, {6, 1}, {7, 1}};
MultipleListPlot[lst1, lst2, lst3];
MultipleListPlot[lst1, lst2, lst3, PlotJoined →True];
```



可以用 Mathematica 画不同类型的条形图,这些命令都包含在软件包 **Graphics`Graphics**

中.

- **BarChart[数据表 1, 数据表 2, …, 选项]** 画单个条形图. 如果指定了多个数据表, 那么根据数据在表中的位置, 把条形图分成彼此邻近的组. 每个数据表的形式为{y1, y2, …}或者{{x1, y1}, {x2, y2}, …}, 而且列表可以由不同数量组成.
- **StackedBarChart[数据表 1, 数据表 2, …, 选项]** 当给出多组数据表时可以使用该函数. 这时同一组中的条形图是堆积在一起, 并不是相邻摆放.
- **PercentileBarChart[数据表 1, 数据表 2, …, 选项]** 生成一个堆积的条形图, 只是这里每一条的高度总和都是 100%.
- **GeneralizedBarChart[数据表 1, 数据表 2, …, 选项]** 生成数据表的条形图, 这里的每个数据表指定条形的位置、高度与宽度.

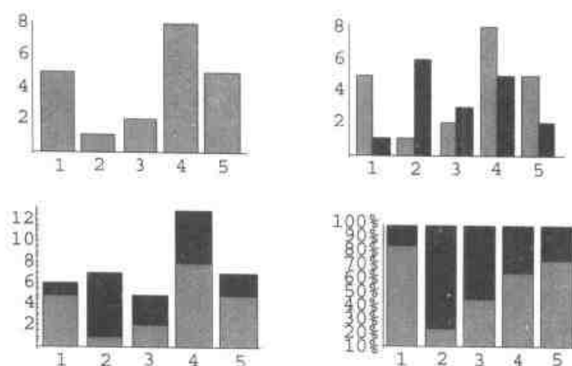
下面给出的是使用 BarChart 时可以使用的选项:

- **BarOrientation →Horizontal** 使条形图从竖直摆放样式变成水平摆放样式. **BarOrientation →Vertical** 为默认值.
- **BarEdges →False** 去掉包围每个条形图的轮廓线.
- **BarValues →True** 给每个条形图标上它所表示的数值.
- **BarLabels →字符串表** 利用条形图位置对应字符串表中的字符串给条形图加上标签.
- **TextStyle** 可以改变在图形文本中所用的默认字体和尺寸. 例如, **TextStyle → {FontFamily →字体名, FontSize →尺寸}** 就是该选项的一个简单而实用的例子.
- **BarSpacing →数值** 指定两个条形图之间的间距, 这里的数值表示占一个条形图总宽度的百分比.
- **BarGroupSpacing →数值** 指定相邻条形图组之间的距离, 这里的数值表示占一个条形图宽度的百分比.
- **BarStyle** 指定每个条形图的样式(如 RGBColor, Hue, GrayLevel, 等等).
- **BarEdgeStyle** 指定条形图边界的样式.

例 33

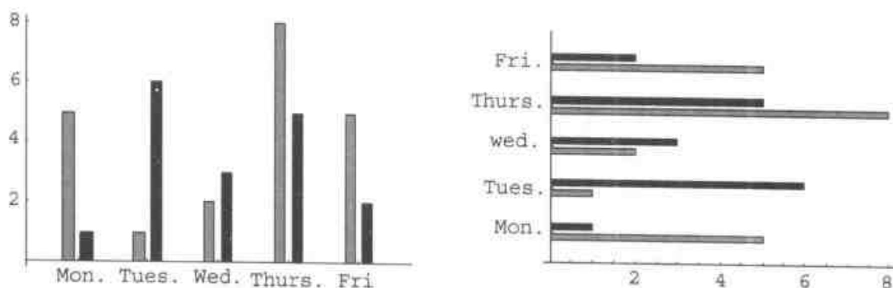
```
<<Graphics`Graphics`
dataset1 = {5, 1, 2, 8, 5};
dataset2 = {1, 6, 3, 5, 2};
g1 = BarChart[dataset1, BarStyle →{GrayLevel[.7]},
     DispalyFunction →Identity];
g2 = BarChart[dataset, dataset2,
     BarStyle →{GrayLevel[.7], GrayLevel[0]},
     DisplayFunction →Identity];
g3 = StackedBarChart[dataset, dataset2,
     BarStyle →{GrayLevel[.7], GrayLevel[0]},
     DisplayFunction →Identity];
g4 = PercentileBarChart[dataset, dataset2,
     BarStyle →{GrayLevel[.7], GrayLevel[0]},
     DisplayFunction →Identity];
g = GraphicsArray[{{g1, g2}, {g3, g4}}];
Show[g, DisplayFunction → $DisplayFunction];
```

BarStyle 的默认值为红与蓝条形图, 在本书此处以及后面例子中则用 GrayLevel.



例 34

```
<<Graphics`Graphics`
dataset1 = {5, 1, 2, 8, 5};
dataset2 = {1, 6, 3, 5, 2};
g1 = BarChart[dataset1, dataset2,
    BarLabels →{"Mon.", "Tues.", "Wed.", "Thurs.", "Fri."},
    BarSpacing →.1, BarGroupSpacing →.5,
    BarStyle →{GrayLevel[.7], GrayLevel[0]},
    DisplayFunction →Identity];
  g2 = BarChart[dataset1, dataset2,
      BarLabels →{"Mon.", "Tues.", "Wed.", "Thurs.", "Fri."},
      BarSpacing →.1, BarGroupSpacing →.5,
      BarOrientation →Horizontal,
      BarStyle →{GrayLevel[.7], GrayLevel[0]},
      DisplayFunction →Identity];
  g = GraphicsArray[{g1, g2}];
  Show[g, DisplayFunction → $DisplayFunction];
```



饼形图可以用 **PieChart** 命令构造出来,这条命令也是包含在软件包 **Graphics`Graphics`** 中.

- **PieChart[{x1, x2, …}]** 把数组{x1, x2, …}中的数据转化为饼形显示,即用扇形表示,每部分的大小正比于 x1, x2, …的值.
- **PieChart[{x1, x2, …}, 选项]** 构造饼形图,并使用各种选项.

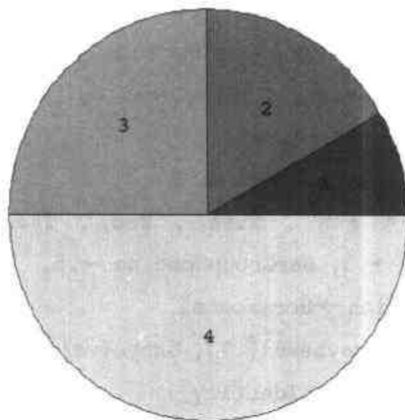
下面给出的是在 PieChart 中可以使用的各种选项:

- **PieLabels →标签表**,其中标签表为一组文本字符串,分别对应于图中楔形的标签.楔形默认的标签为连续整数 1, 2, ….**PieLabels →None** 使得不显示标签.
- **PieExploded →All**.使得饼形图有突出效果.如果不想突出所有的楔形,那么 **Pie-**

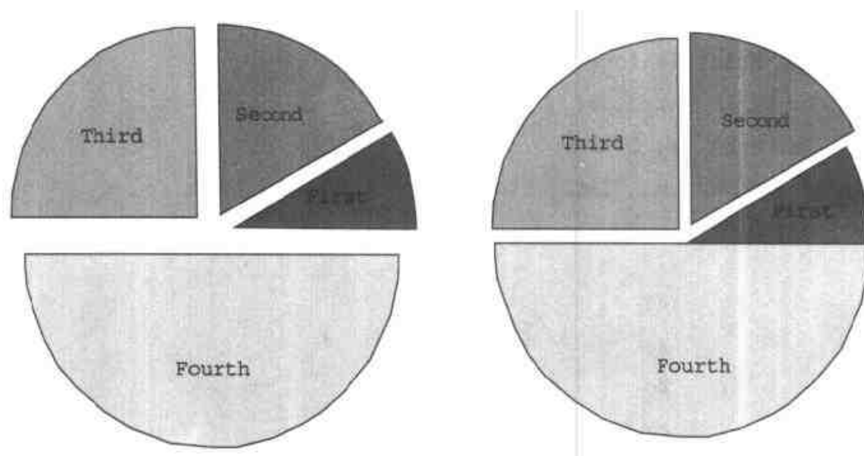
Exploded →{楔形 1, 楔形 2, …}使得只有指定的楔形才会被突出.

- **PieStyle →样式表**,其中样式表是一组图形样式(如 RGBColor, Hue, GrayLevel 等等.),从而可以使饼形图中每个楔形具有各种不同的样式.如果楔形数目多于样式数目,PieStyle 要被循环使用.
- **PieLineStyle →直线样式表**,其中直线样式表为一组直线样式,利用这个选项可以给楔形的边界线指定不同的样式.如果楔形数目多于样式数目,PieLineStyle 要被循环使用.

例 35
```
<<Graphics`Graphics`
datalist = {1.5, 3, 4.5, 9};
PieChart[datalist, PieStyle ->
{GrayLevel[.6], GrayLevel[.7], GrayLevel[.8], GrayLevel[.9]}];
```



例 36
```
<<Graphics`Graphics`
datalist = {1.5, 3, 4.5, 9};
labellist = {"First", "Second", "Third", "Fourth"};
g1 = PieChart[datalist,
     PieLables ->labellist,
     PieExploded ->All,
     PieStyle ->{GrayLevel[.6], GrayLevel[.7],
                 GrayLevel[.8], GrayLevel[.9]},
     DisplayFunction ->Identity];
g2 = PieChart[datalist,
     PieLables ->labellist,
     PieExploded ->{2, 3},          ←只有第二个与第三个楔形被突出显示.
     PieStyle ->{GrayLevel[.6], GrayLevel[.7],
                 GrayLevel[.8], GrayLevel[.9]},
     DisplayFunction ->Identity];
g = GraphicsArray[{g1, g2}];
Show[g, DisplayFunction -> $DisplayFunction];
```

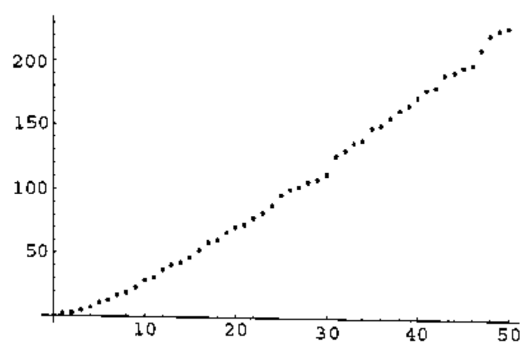


习题解答

4.18 画出前 50 个素数的图形.

解 ☞

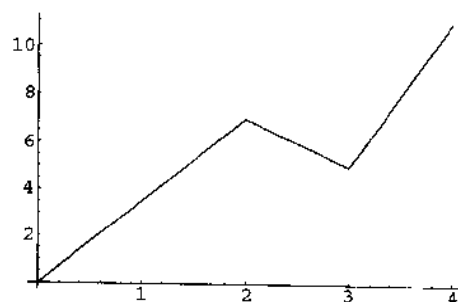
```
primelist = Table[Prime[k], {k, 1, 50}];
ListPlot[primelist];
```



4.19 描出点 (0, 0), (2, 7), (3, 5) 与 (4, 11),并用直线段把它们连接起来.

解 ☞

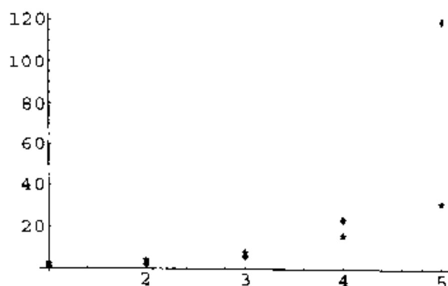
```
lst = {{0, 0}, {2, 7}, {3, 5}, {4, 11}};
ListPlot[lst, PlotJoined →True];
```



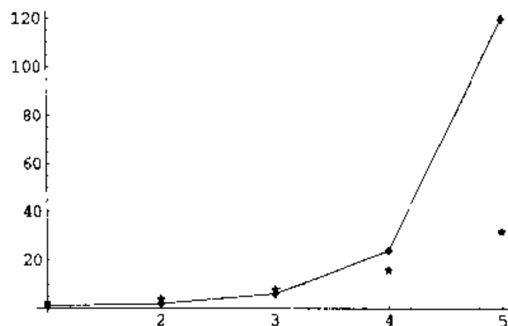
4.20 画出对应于 $n!$ 与 2^n, $n = 1, 2, 3, 4, 5$ 时的点.首先只是画出点,然后再把它们连接起来.

解 ☞

```
<<Graphics`MultipleListPlot`
lst1 = Table[n!, {n, 1, 5}];
lst2 = Table[2^n, {n, 1, 5}];
MultipleListPlot[lst1, lst2, PlotRange->All];
```




```
MultipleListPlot[lst1, lst2, PlotRange->All, PlotJoined->True];
```



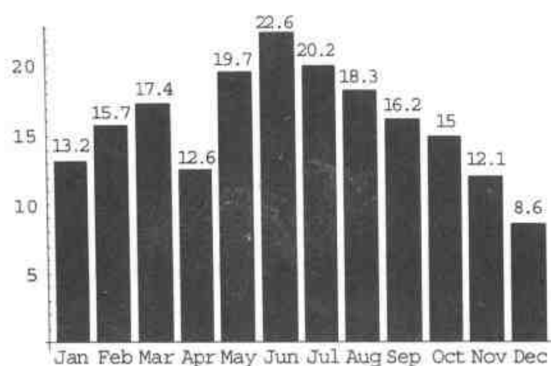
4.21 某公司的月度销售额(以千美元为单位)为

一月	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月
13.2	15.7	17.4	12.6	19.7	22.6	20.2	18.3	16.2	15.0	12.1	8.6

构造一个条形图演示这组数据.

解 ☞

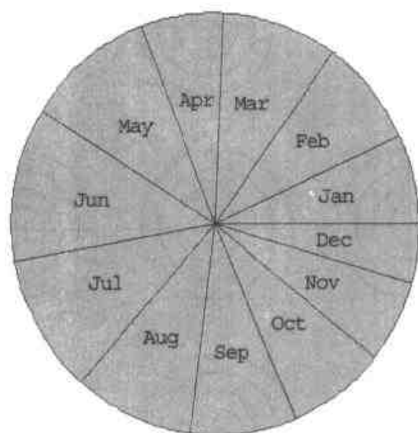
```
<<Graphics`Graphics`
months = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
        "Sep", "Oct", "Nov", "Dec"};
salesdata = {13.2, 15.7, 17.4, 12.6, 19.7, 22.6, 20.2, 18.3,
            16.2, 15.0, 12.1, 8.6};
BarChart[salesdata,
   BarLabels->months,
   BarValues->True,
   PlotRange->All,
   TextStyle->{FontFamily->Helvetica, FontSize->9}];
```



4.22 用一个单色饼图演示前一个习题中的数据.

解 ☞

```
<<Graphics`Graphics`
months = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
        "Sep", "Oct", "Nov", "Dec"};
salesdata = {13.2, 15.7, 17.4, 12.6, 19.7, 22.6, 20.2, 18.3,
            16.2, 15.0, 12.1, 8.6};
PieChart[salesdata, PieLabes->months,
                   PieStyle->{GrayLevel[.8]}];
```



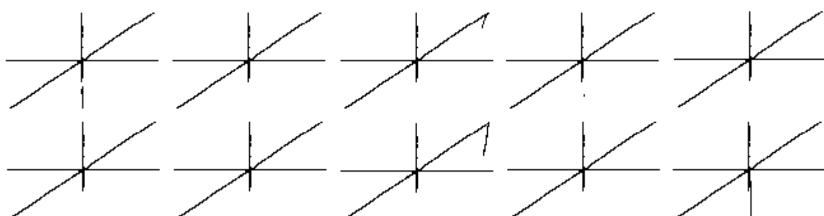
4.4 动画

利用 **Animate Selected Graphics** 选项(可在 **Cell** 菜单中找到)就可以简单而快捷地产生实用动画效果.这个选项快速地连续演示几种不同的图像,从而造成运动的假象.

创建动画中的第一步就是构造一系列图像,每幅图像都稍有些不同.(拥有的图像数目越大,相邻图像之间的差别越小,那么就会看到越光滑的动画效果.)接着用鼠标选择图像单元括号.最后选择 **Cell ⇒Animate Selected Graphcis**, Mathematica 就会为你做余下的事情了.

下面的例子演示了在 x^n 中当奇数 n 变大时的很有趣的动画效果.(Mathematica 会把图形上下竖直摆放.)

例 37
```
Do[Plot[xk, {x, -1, 1},
    PlotRange->{-1, 1}, Ticks->False], {k, 1, 19, 2}]
```



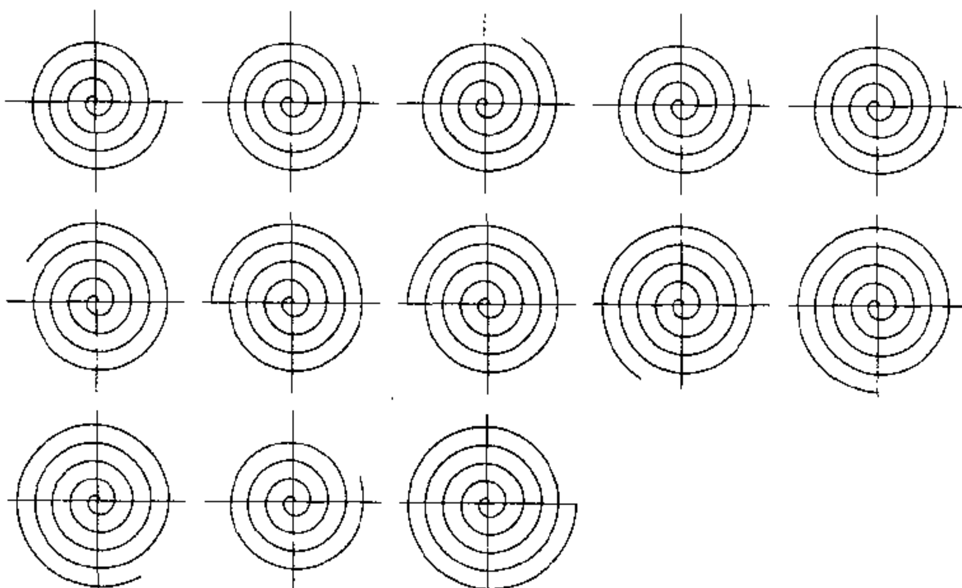
选择这些图像的单元括号,然后再选择 **Cell ⇒Animate Selected Graphcis**,就可以看到动画效果了.

习题解答

4.23　构造阿基米德螺线 $r=\theta$, θ 从 8π 变到 10π 的动画.

解 ☞

```
<<Graphics`Graphics`
Do[PolarPlot[θ, {θ, 0, 8π + φ}, Ticks →False,
  PlotRange →{{-10π, 10π}, {-10π,10π}}], {φ,0,2π,π/6}]
```



选择这些图像的单元括号,然后再选择 **Cell ⇒Animate Selected Graphcis**,这样螺线就会旋转起来.

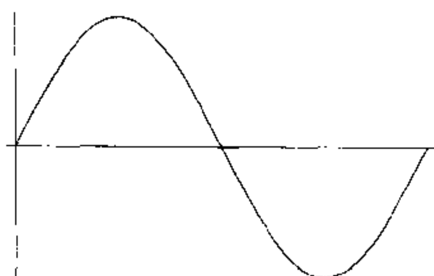
4.24　构造点沿从 0 到 2π 的正弦曲线移动的动画.

解 ☞

首先构造正弦曲线.

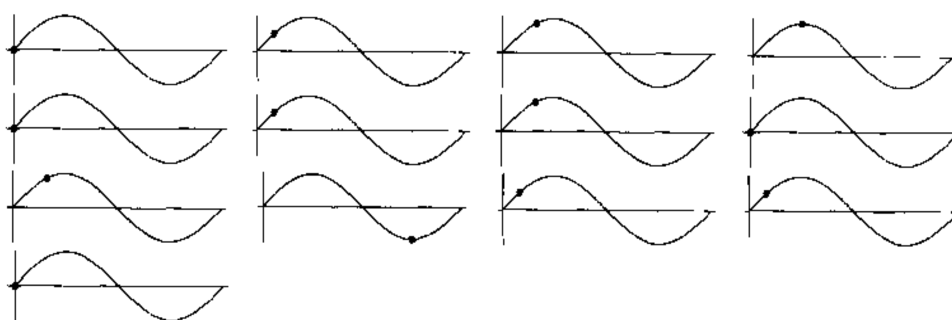
```
sincurve = Plot[Sin[x], {x, 0, 2π}, Ticks →False];
```

现在构造一系列半径为 0.05 的圆盘表示的点.



```
Do[Show[sincurve, Graphics[Disk[{x, Sin[x]}, 0.05]],
        PlotRange→{{0, 2π}, {-1, 1}},
        AspectRatio→Automatic], {x,0,2π,π/6}]
```

这样就得到了下述的图形序列：



选择这些图像的单元格括号,然后再选择 **Cell ⇒Animate Selected Graphcis**,点就会沿曲线移动起来.

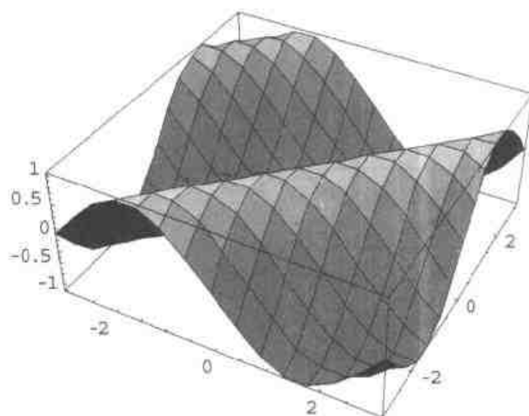
第五章　三 维 图 形

5.1　绘制二元函数的图形

具有两个变量的函数可以看作是三维空间中的曲面.绘制曲面的最简单命令是 **Plot3D**.

■ **Plot3D[f[x, y], {x, xmin, xmax}, {y, ymin, ymax}]** 绘制函数 f[x, y]在矩形 xmin≤x≤xmax, ymin≤y≤ymax 上定义的三维曲面.

例 1　**Plot3D[Sin[x - y], {x, - π, π}, {y, - π, π}];**



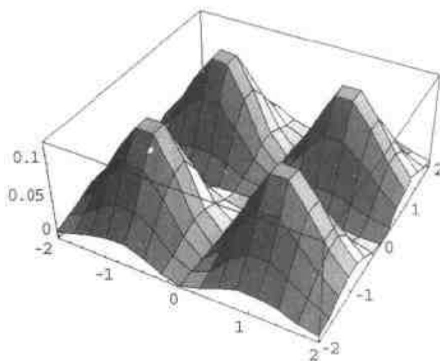
PlotPoints 选项指定生成图形时在每个方向上所用的点数.与二维图形不同,三维图形的默认值为 **PlotPoints →15**. 这种设置有时候会使图形显得不光滑.增大 PlotPoints 的值就会缓解这种现象.

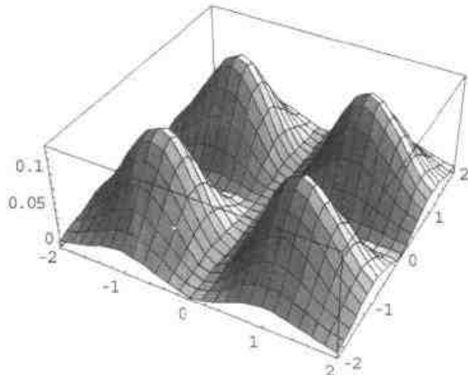
- **PlotPoints →n** 指定在每个方向上使用 n 个取样点.
- **PlotPoints →{nx, ny}** 指定在 x 轴与 y 轴方向上分别使用 **nx** 与 **ny** 个取样点.

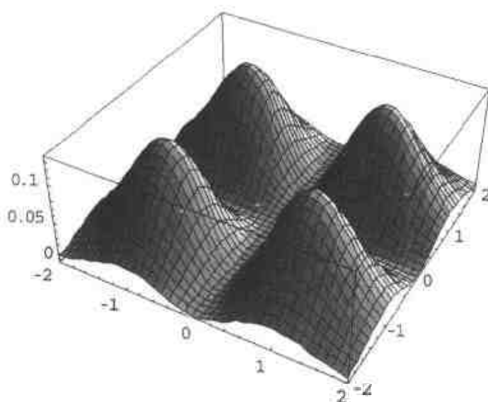
下面这个例子说明了增大 PlotPoints 的值是如何改变所得图形的"光滑性"的.

例 2

```
f[x_, y_] = x² y² Exp[ - (x² + y²)];
Plot3D[f[x, y], {x, - 2,2}, {y, - 2,2}];
Plot3D[f[x, y], {x, - 2,2}, {y, - 2,2}, PlotPoints →25];
Plot3D[f[x, y], {x, - 2,2}, {y, - 2,2}, PlotPoints →40];
```







绝大多数的二维图形选项在 **Plot3D** 中都可以使用，除此而外，它还具有几个选项．最常用的是：

- **Axes→False** 不显示坐标轴．
- **Axes→{True 或 False, True 或 False, True 或 False}**，其中的 **True 或 False**，就是指要么为 True，要么为 False，以控制每个坐标轴显示与否．
- **Boxed→False** 不显示包围图形的盒子(包围盒)．
- **BoxRatios→{sx, sy, sz}** 指定三维图形包围盒各边长度的比率． • **Ticks→False** 不显示所有坐标轴上的刻度以及相应的标签．
- **Ticks→{True 或 False, True 或 False, True 或 False}**，其中的 **True 或 False**，就是指要么为 True，要么为 False，以控制每个坐标轴是否显示刻度．

FaceGrids 选项确定是否绘制在包围盒的每个侧面上的格线．

- **FaceGrids→All** 在包围盒的所有六个侧面上都显示格线．
- **FaceGrids→None**（默认值）不绘制任何格线．
- **FaceGrids→{{x1, y1, z1}, {x2, y2, z2}, …, {x6, y6, z6}}** 使得每个侧面具有不同的格线．其中每个列表的三个数中必须有两个为零，第三个为 ±1 以指定六个侧面中的一个．

AxesEdge 选项指定在包围盒上的哪个边界上的坐标轴被显示出来．

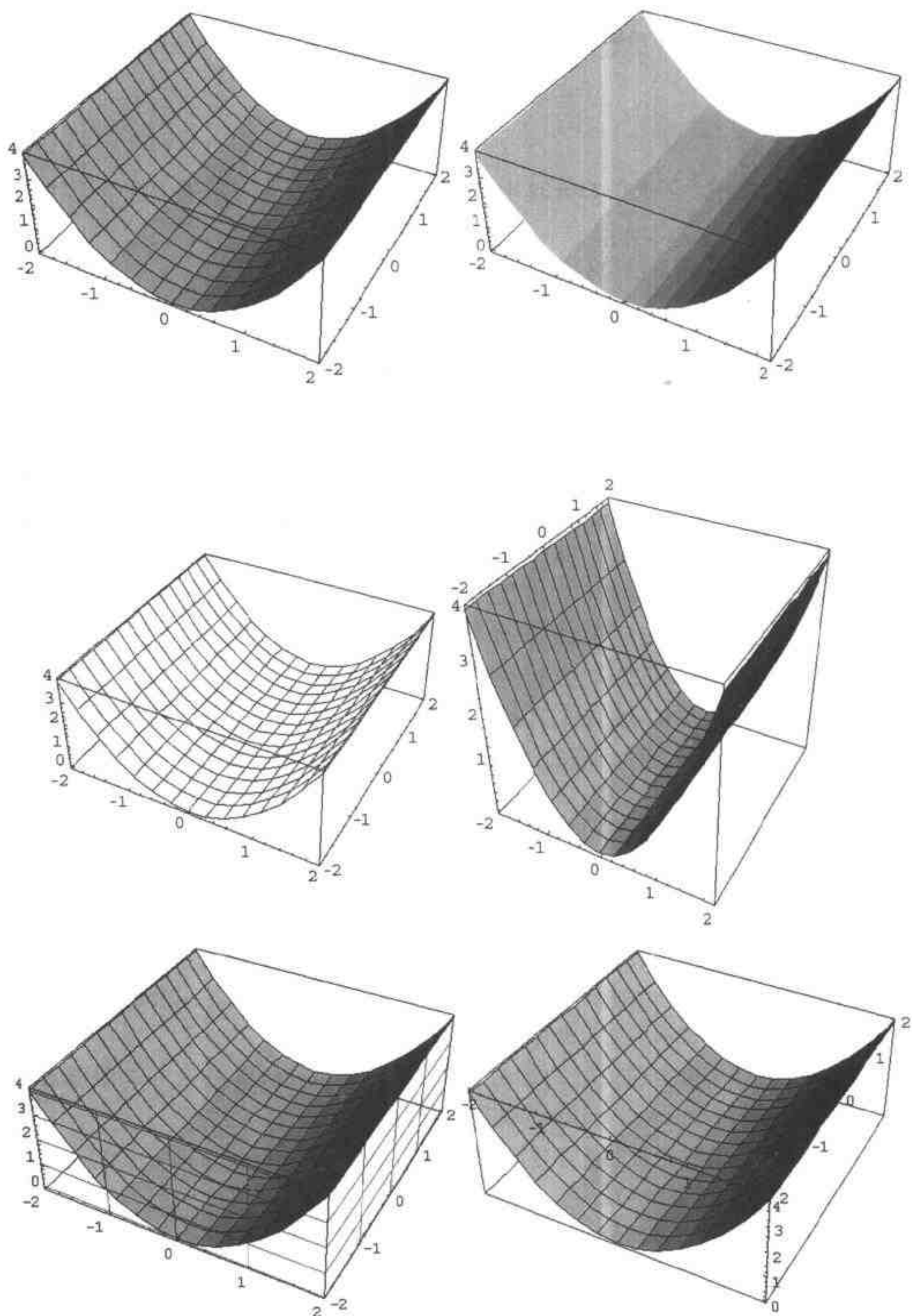
- **AxesEdge→Automatio**（默认值）由 Mathematica 确定显示哪个边界轴．
- **AxesEdge→{{y_1, z_1}, {x_2, z_2}, {x_3, y_3}}**，其中 x, y, z 的每个值为 1 或 −1，指定显示哪个边界轴．1 表示在边界上按最大坐标值显示坐标轴；−1 则表示最小坐标值．三个列表中任一个都可以用 **Automatic**(表示由 Mathematica 确定是否显示坐标轴)或者 **None**(表示不显示坐标轴)代替．
- **BoxStyle** 选项指定包围盒的绘制方式．可以给 **BoxStyle** 指定一组图形指令，如 Dashing, Thickness, GrayLevel 或者 RGBColor．
- **Mesh** 选项确定在曲面上是否显示格线．缺省值为 **Mesh→True**．如果设置为 **Mesh→False** 就会去掉格线．
- **Shading** 选项确定是否给曲面加阴影效果．缺省值为 **Shading→True**．

下面的例子用几种不同的方式绘制抛物柱面 $z = x^2$．

例 3
```
Plot3D[x², {x, -2, 2}, {y, -2, 2}];
Plot3D[x², {x, -2, 2}, {y, -2, 2}, Mesh→False];
Plot3D[x², {x, -2, 2}, {y, -2, 2}, Shading→False];
Plot3D[x², {x, -2, 2}, {y, -2, 2}, BoxRatios→{1, 1, 1}];
Plot3D[x², {x, -2, 2}, {y, -2, 2}, FaceGrids→{{1,0,0}, {0, -1,0}}];
```

Plot3D[x², {x, -2, 2}, {y, -2, 2}, AxesEdge→{{-1,1},{1,1},{1,-1}}];

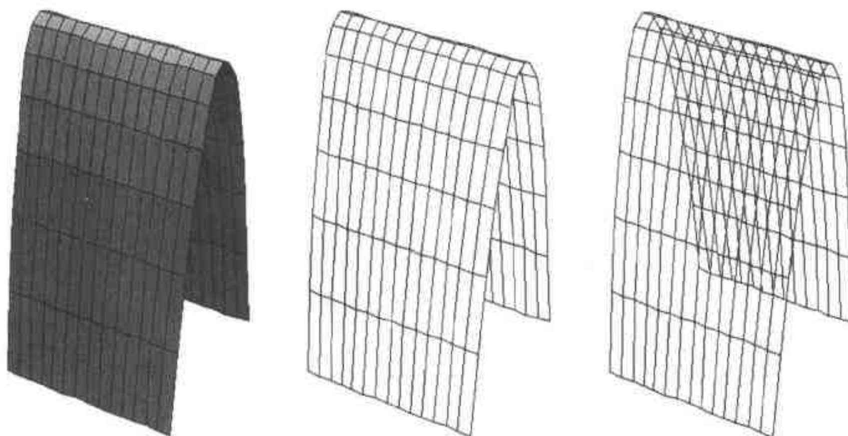
在三维图形中，为了得到令人满意的立体效果，首先生成一系列的多边形，然后加上阴影．虽然选项 Shading→False 使得不进行阴影处理，但是绘制的多边形是不透明的，因而在某个曲面后面的曲面是看不到的．选项 **HiddenSurface** 控制是否把多边形绘成不透明的或透明的．



- **HiddenSurface →False** 把多边形绘成透明的（即只画出连接线段），从而所有的曲面都可见．缺省值 **HiddenSurface →True** 生成不透明多边形．

例 4　Plot3D[1 - y², {x, -5, 5}, {y, -5, 5}, BoxRatios→{1, 1, 2},

```
        Boxed →False, Axes →False];
Plot3D[1 - y^2, {x, -5, 5}, {y, -5, 5}, BoxRatios →{1, 1, 2},
        Boxed →False, Axes →False, Shading →False];
Plot3D[1 - y^2, {x, -5, 5}, {y, -5, 5}, BoxRatios →{1, 1, 2},
        Boxed →False, Axes →False, HiddenSurface →False];
```



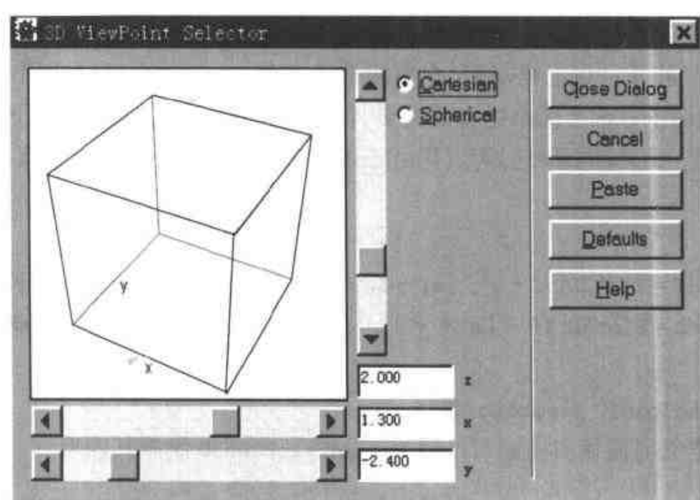
有许多不同的方法查看三维图形.选项 **ViewPoint** 指定在包含图形的包围盒外面的一个固定点,以从该点查看图形.

- **ViewPoint →{x, y, z}** 给出相对于要绘制曲面的包围盒中心的视点位置.

视点坐标要经过放缩,使得包围盒的最长边为 1. 视点必须在包围盒的外面.一般地,所选择的视点离曲面越远,曲面的变形就越轻.

ViewPoint 参数的默认值为{1.3, -2.4, 2.0}. ViewPoint→{0, -2, 0}就会直接从曲面的前面查看.{0, 0, 2}直接从上面查看曲面.

在 Mathematica 中有一种简单的方法确定 ViewPoint 的参数.选择 **Input ⇒3D ViewPoint Selector** …就会进入一个交互式确定视点的界面.

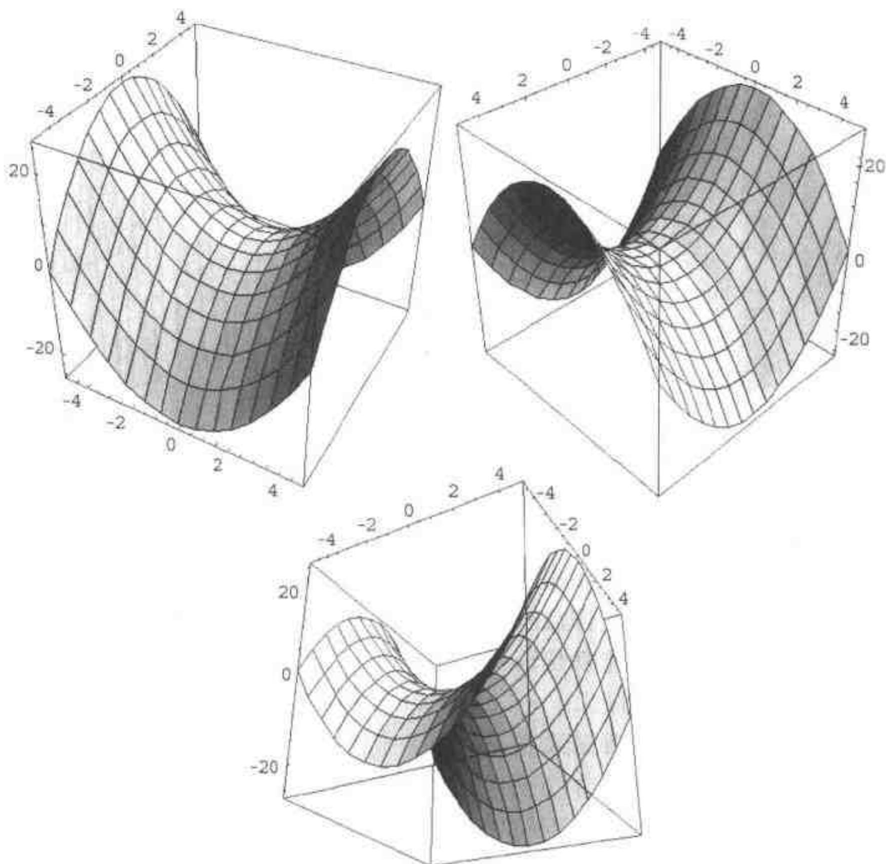


在这个窗口中可以直接调整 x, y, z 坐标,或者利用鼠标拖动方框.当调整结束时,点击"Paste"就会在当前光标位置插入文本

ViewPoint →{1.300, -2.400, 2.000}.

下面的例子演示了从不同的视点查看双曲抛物面 $z = x^2 - y^2$ 的结果.

例 5
```
Plot3D[x^2 - y^2, {x, -5, 5}, {y, -5, 5}, BoxRatios→{1, 1, 1}];
Plot3D[x^2 - y^2, {x, -5, 5}, {y, -5, 5}, BoxRatios→{1, 1, 1},
                    ViewPoint→{2, 2, 2}];
Plot3D[x^2 - y^2, {x, -5, 5}, {y, -5, 5}, BoxRatios→{1, 1, 1},
                    ViewPoint→{1.5, -2.6, -1.5}];
```

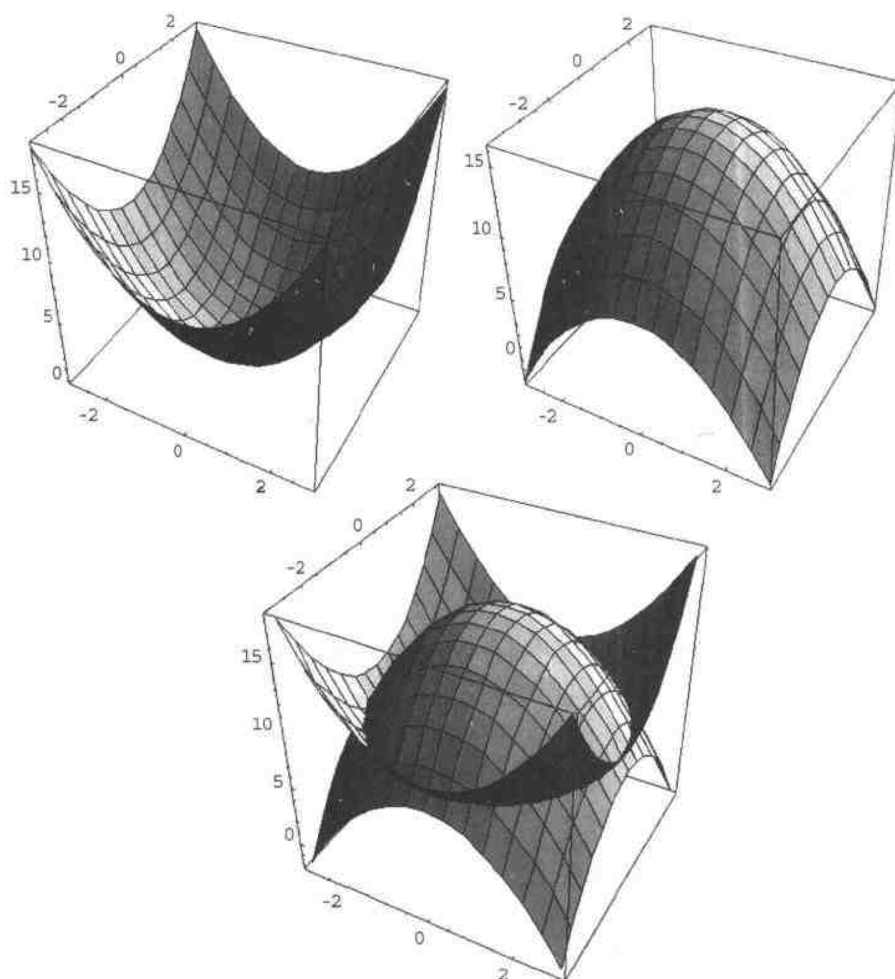


Plot3D 一次只能画一张三维曲面. 在第四章讨论的 Show 命令可以在同一坐标系中显示几张曲面. 下面的例子显示了两个抛物面的交. 首先单独显示曲面, 然后为了比较, 在一个包围盒中同时显示它们. 如果希望的话, 可以用 DisplayFunction→Identity 选项不显示前两个图形.

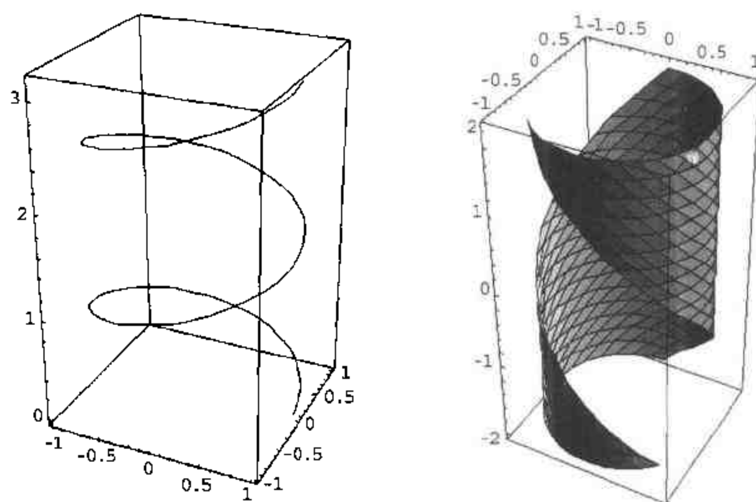
例 6
```
paraboloid1 = Plot3D[x^2 + y^2, {x, -3,3}, {y, -3,3}, BoxRatios→{1, 1, 1}];
paraboloid2 = Plot3D[16 - (x^2 + y^2), {x, -3, 3}, {y, -3, 3}, BoxRatios→{1, 1,
1}];
Show[paraboloid1, paraboloid2];
```

由参数方程定义的曲线和曲面可以用 **ParametricPlot3D** 绘制其图形.

- **ParametricPlot3D[{x[t], y[t], z[t]}, {t, tmin, tmax}]** 在三维空间中绘制 tmin≤t≤tmax 确定的空间曲线.
- **ParametricPlot3D[{x[s, t], y[s, t], z[s, t]}, {s, smin, smax}, {t, tmin, tmax}]** 在三维空间中绘制 smin≤s≤smax 与 tmin≤t≤tmax 确定的曲面.



例 7 **ParametricPlot3D[{Cos[t], Sin[t], t/4}, {t, 0, 4π},**
ViewPoint →{1.292, -2.788, 1.418}];
ParametricPlot3D[{Sin[s+t], Cos[s+t], s}, {s, -2, 2}, {t, -2, 2}];



Plot3D 显示在直角坐标系中的曲面. 而在包含圆柱面、球面和圆锥面的问题中, 使用柱面

坐标系和球面坐标系就相当方便.

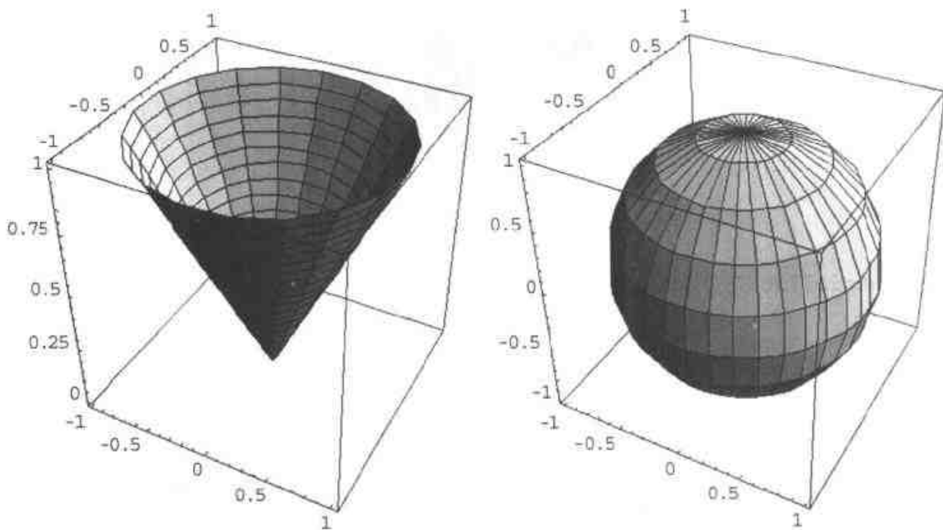
点 $P(x, y, z)$ 的柱面坐标为 (r, θ, z)，其中 $x = r\cos\theta$, $y = \sin\theta$；球面坐标为 (ρ, θ, ϕ)，其中 $x = \rho\sin\phi\cos\theta$, $y = \rho\sin\phi\sin\theta$, $z = \rho\cos\phi$. 这里 r 与 θ 为点 P 在 $x-y$ 平面上投影的极坐标，而 ϕ 为 z 轴和连接 P 与原点的线段所形成的夹角. Mathematica 命令 **CylindricalPlot3D** 与 **SphericalPlot3D** 可以绘制在这些坐标系中定义的曲面. 这两条命令包含在 **Graphics`ParametricPlot3D`** 中，可以在命令中使用所有的三维绘图选项.

特别注释 当遇到球面坐标时，Mathematica 的约定与在许多标准微积分教科书中的约定不同，它交换了 θ 与 ϕ 的位置. 在下面描述的 **SphericalPlot 3D** 命令中虽然与 Mathematica 帮助文件中的描述不同，但它遵从的还是这些约定.

- **CylindricalPlot3D[z[r, θ], {r, rmin, rmax}, {θ, θmin, θmax}]** 生成在柱面坐标中描述的曲面 $z = z(r, \theta)$.
- **SphericalPlot3D[ρ[φ, θ], {φ, φmin, φmax}, {θ, θmin, θmax}]** 生成在球面坐标中描述的曲面 $\rho = \rho(\phi, \theta)$.

例 8 在柱面坐标系中，方程 $z = r$ 表示圆锥面，而在球面坐标系中，$\rho = 1$ 表示单位球面.

```
<<Graphics`ParametricPlot3D`
CylindricalPlot3D[r, {r, 0, 1}, {t, 0, 2π}, BoxRatios→{1, 1, 1}];
SphericalPlot3D[1, {t, 0, 2π}, {ρ, 0, π}, BoxRatios→{1, 1, 1}];
```



习题解答

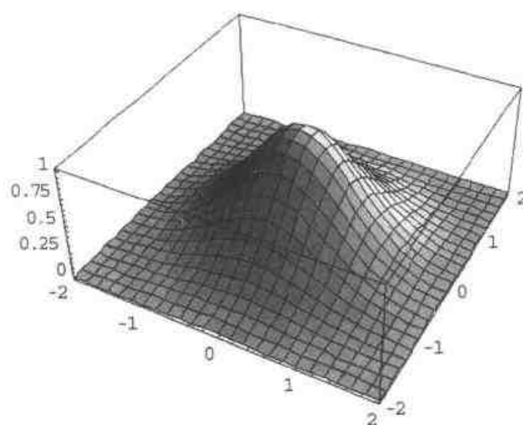
5.1 在定义域 $-2\leqslant x\leqslant 2$ 与 $-2\leqslant y\leqslant 2$ 上绘制函数 $e^{-x^2-y^2}$ 的图形. 注意使用足够多的点，以得到“光滑”的曲面.

解 ☞
```
Plot3D[Exp[-x^2-y^2], {x, -2, 2}, {y, -2, 2}, PlotPoints→25];
```

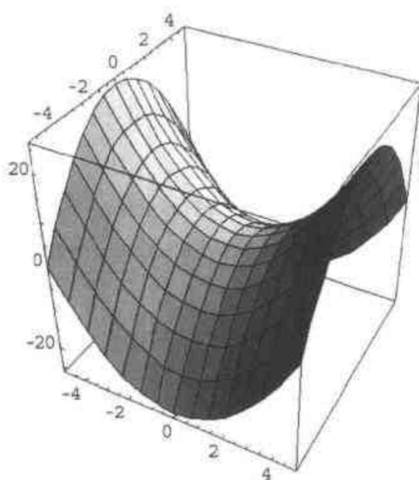
5.2 在立方体盒子中绘出马鞍面 $z = x^2 - y^2$, $-5\leqslant x\leqslant 5$, $-5\leqslant y\leqslant 5$ 的图形. 给出两张图形，一张有格线，另一张没有格线.

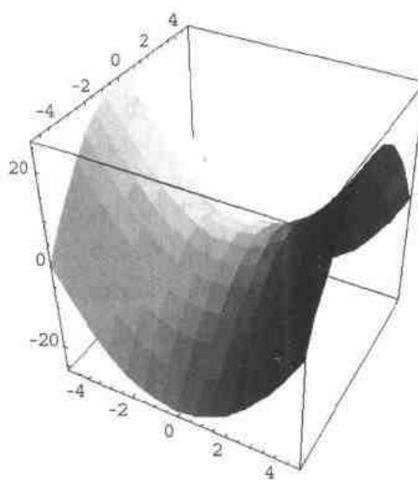
解 ☞

```
Plot3D[x^2-y^2, {x, -5, 5}, {y, -5, 5}, BoxRatios→{1, 1, 1}];
```




```
Plot3D[x^2 - y^2, {x, -5, 5}, {y, -5, 5}, Mesh→False,
        BoxRatios→{1, 1, 1}];
```

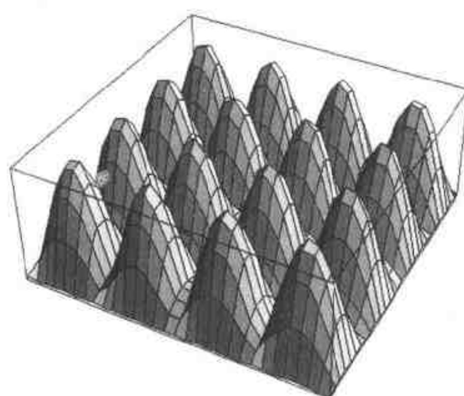




5.3 画出函数 $f(x, y) = |\sin x \sin y|$ 在 $-2\pi \leqslant x \leqslant 2\pi$, $-2\pi \leqslant y \leqslant 2\pi$ 上的图形.不显示坐标轴,使用足够多的点,以得到"光滑的"曲面.

解 ☞

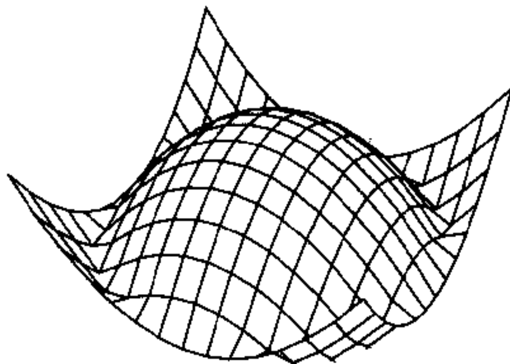
```
Plot3D[Abs[Sin[x] Sin[y]], {x, -2π, 2π}, {y, -2π, 2π},
    Axes→False, PlotPoints→30];
```



5.4 画出曲面 $z=|1-x^2-y^2|$ 在 $-1\leqslant x\leqslant 1$, $-1\leqslant y\leqslant 1$ 上的不加阴影的图形.不显示坐标轴和包围盒.

解 ☞

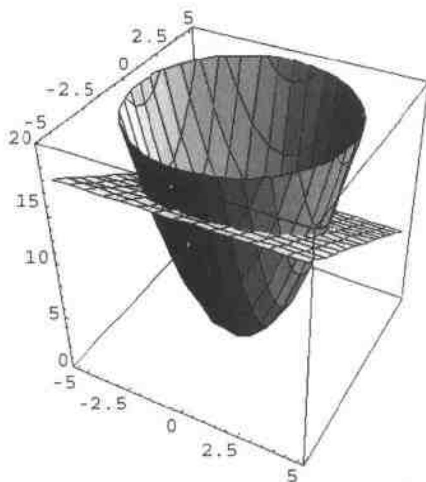
```
Plot3D[Abs[1 - x^2 - y^2], {x, - 1,1}, {y, - 1,1}, Shading→False, Axes→False,
Boxed→False];
```

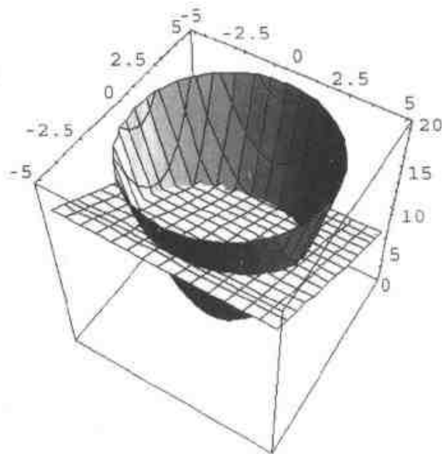


5.5 画出抛物面 $z=x^2+y^2$ 与平面 $y+z=12$ 的交.给出两种不同的视图.

解 ☞

```
paraboloid = Plot3D[x^2 + y^2, {x, - 5,5},{y, - 5,5},DisplayFunction→Identity];
plane = Plot3D[12 - y, {x, - 5, 5}, {y, - 5, 5}, DisplayFunction→Identity];
Show[paraboloid, plane, BoxRatios→{1, 1, 1},
    PlotRange→{0, 20}, DisplayFunction→ $DisplayFunction];
Show[paraboloid, plane, BoxRatios→{1, 1, 1},
    ViewPoint→{1.126, - 1.800, 2.634},
    PlotRange→{0, 20}, DisplayFunction→ $DisplayFunction];
```





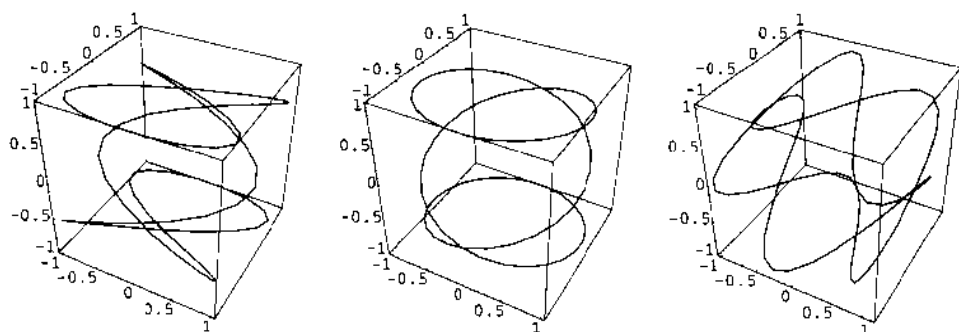
5.6 画出由参数方程 $\begin{cases} x=\cos at, \\ y=\sin bt, \\ z=\sin ct, \end{cases}$ $0\leqslant t\leqslant 2\pi$ 定义的曲线在下述几种情况中的图形:

(i) $a=5, b=3, c=1$; (ii) $a=3, b=3, c=1$; (iii) $a=2, b=5, c=2$.

(这些曲线称为利萨如曲线.)

解 ☞

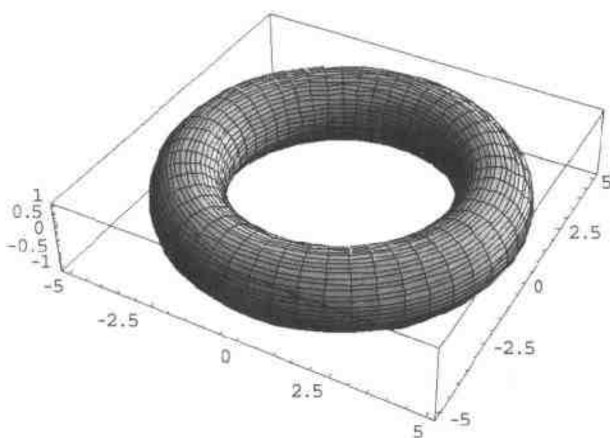
```
g1 = ParametricPlot3D[{Cos[5t], Sin[3t], Sin[t]}, {t, 0, 2π},
                     DisplayFunction →Identity];
g2 = ParametricPlot3D[{Cos[3t], Sin[3t], Sin[t]}, {t, 0, 2π},
                     DisplayFunction →Identity];
g3 = ParametricPlot3D[{Cos[2t], Sin[5t], Sin[2t]}, {t, 0, 2π},
                     DisplayFunction →Identity];
g = GraphicsArray[{g1, g2, g3}]
Show[g, DisplayFunction → $DisplayFunction];
```



5.7 画出由$\begin{cases}x=(4+\sin s)\cos t,\\ y=(4+\sin s)\sin t,\ 0\leqslant s,\ t\leqslant 2\pi\\ z=\cos s,\end{cases}$定义的圆环面的图形.

解 ☞

```
x[s_, t_] = (4 + Sin[s]) Cos[t];
y[s_, t_] = (4 + Sin[s]) Sin[t];
z[s_, t_] = Cos[s];
ParametricPlot3D[{x[s, t], y[s, t], z[s, t]}, {s, 0, 2π},
                {t, 0, 2π}, PlotPoints →50];
```

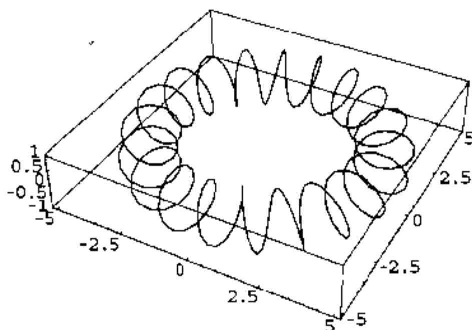


5.8 画出空间曲线$\begin{cases}x=(4+\sin 20t)\cos t,\\ y=(4+\sin 20t)\sin t,\ 0\leqslant t\leqslant 2\pi\\ z=\cos 20t,\end{cases}$的图形.这条曲线称为圆环螺线,因为它

位于圆环面上(见习题 5.7).

解 ☞

```
x[t_] = (4 + Sin[20t]) Cos[t];
y[t_] = (4 + Sin[20t]) Sin[t];
z[t_] = Cos[20t];
ParametricPlot3D[{x[t], y[t], z[t]}, {t, 0, 2π}, PlotPoints →250];
```

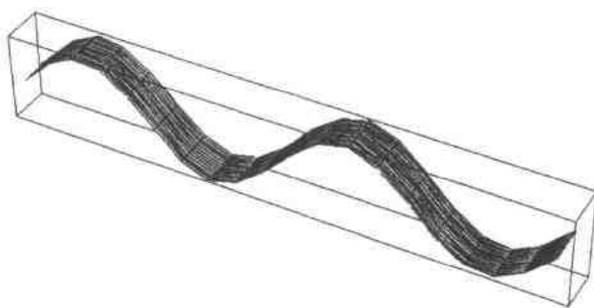


5.9 画出具有一个单位宽,形状为从 0 到 4π 的正弦曲线的"带子"图形.

解 ☞

这张曲面的参数表示为$\begin{cases} x = t \\ y = s, \quad 0 \leqslant s \leqslant 1, 0 \leqslant t \leqslant 4\pi. \\ z = \sin t, \end{cases}$

```
ParametricPlot3D[{t, s, Sin[t]}, {s, 0, 1}, {t, 0, 4π},Axes →False];
```

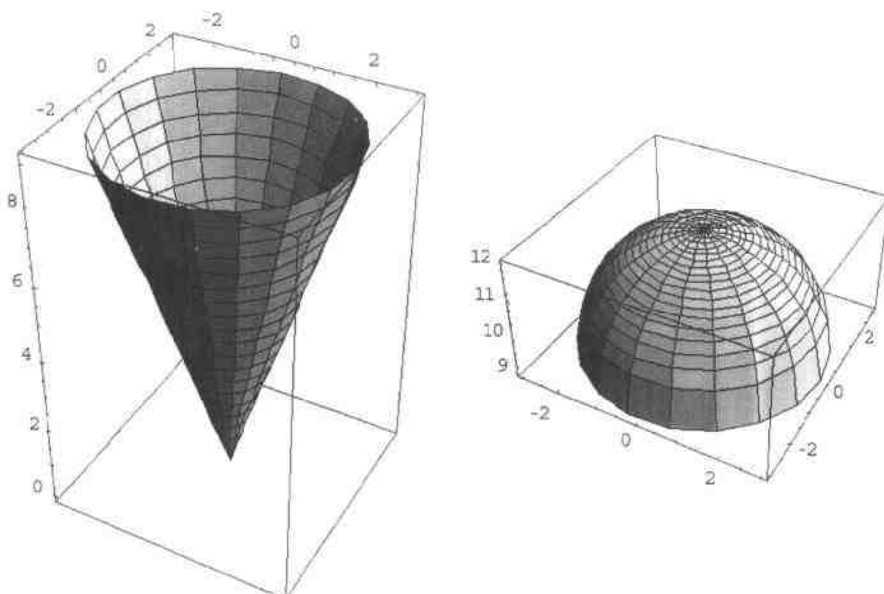


5.10 利用柱面坐标画出"冰淇淋锥"即由锥面 $z = 3\sqrt{x^2 + y^2}$ 以及球面 $x^2 + y^2 + (z - 9)^2 = 9$ 的上半部分所构成几何体的图形.

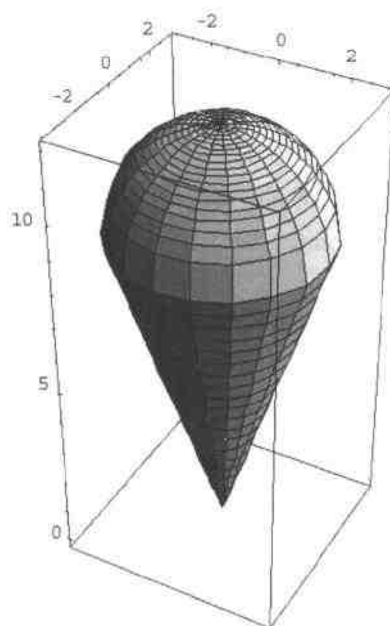
解 ☞

在柱面坐标系中,这个圆锥的方程为 $z = 3r$,半球面方程为 $z = 9 + \sqrt{9 - r^2}$.

```
<<Graphics`ParametricPlot3D`
cone = CylindricalPlot3D[3r, {r, 0, 3}, {θ, 0, 2π},
                DisplayFunction →Identity];
    hemisphere = CylindricalPlot3D[9 + √(9 - r²), {r, 0, 3},{θ,0,2π},
                DisplayFunction →Identity];
g = GraphicsArray[{cone, hemisphere}];
Show[g, DisplayFunction → $DisplayFunction];
```




```
Show[cone, hemisphere, DisplayFunction → $DisplayFunction];
```

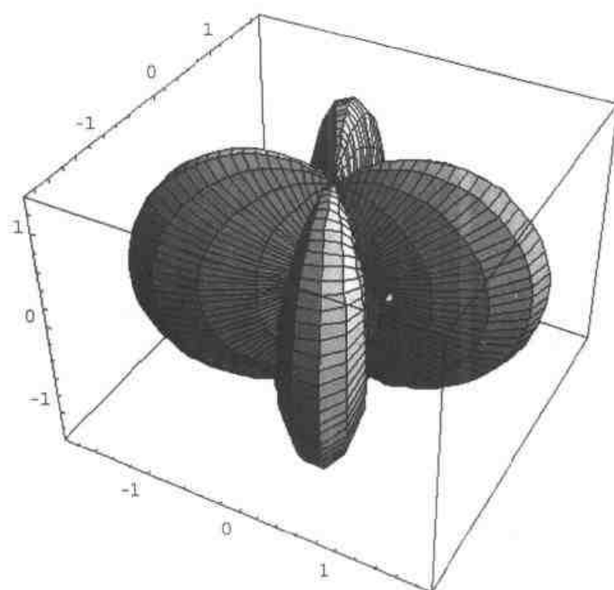


5.11 画出在球面坐标系中下述方程定义的曲面的图形：

$$\rho = 1 + \sin 4\theta \sin\phi, 0 \leqslant \theta \leqslant 2\pi, 0 \leqslant \phi \leqslant \pi.$$

解 ☞

```
<<Graphics`ParametricPlot3D`
SphericalPlot3D [1 + Sin[4θ]Sin[ϕ], {ϕ, 0, π}, {θ, 0, 2π}, PlotPoints → 50];
```



5.2 其它的绘图命令

二元函数 $f(x, y)$ 的层次曲线就是由 $f(x, y)=k$ 确定的一幅二维图形,这里 k 为某个指定值.所谓轮廓图就是在同一坐标系中画出的层次曲线集合.

Mathematica 命令 **ContourPlot** 绘制二元函数的轮廓图.在轮廓图中把具有相同高度的点连接起来.缺省情况中轮廓是对应于函数的一系列等间距值的.

■ **ContourPlot[f[x, y], {x, xmin, xmax}, {y, ymin, ymax}]** 绘制函数 $f(x, y)$在由 xmin, xmax, ymin 与 ymax 确定的矩形上的轮廓图.

由 Mathematica 生成的轮廓图是有阴影的,从而使得 $f(x, y)$的值越高的区域,显示得就越淡.与所有的 Mathematica 图形命令一样,可以利用选项控制图形的外观.

- **Contours →n** 确定要绘制的轮廓数.缺省情况下是绘制 10 条等间距的曲线.
- **Contours →{k1, k2, …}** 绘制对应于函数值 k1, k2, …的轮廓线.
- **ContourShading →False** 不显示阴影.如果所用的显示器或打印机不能很好地处理灰度时,这个选项是相当有用的.
- **ContourLines →False** 去掉分开轮廓的线段.
- **PlotPoints →n** 控制构造曲线的自适应算法在每个方向上要使用的点数.默认值为 15.(二维图形的标准默认值为 25.)

可以利用命令 **Options[ContourPlot]**得到选项的完整清单.

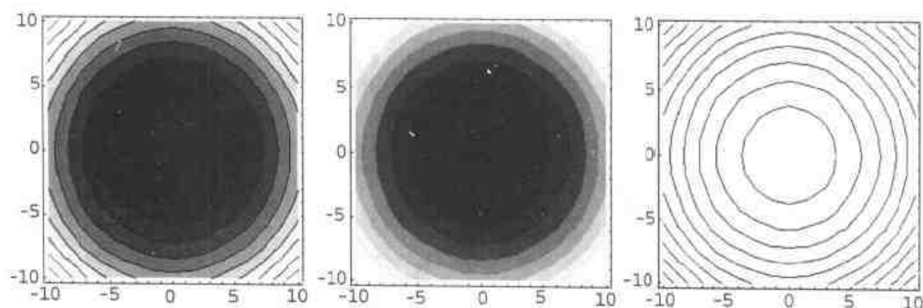
例 9 给出抛物面 $z=x^2+y^2$ 的轮廓图.注意这时的层次曲线为圆 $x^2+y^2=k$.

```
ContourPlot[x^2 + y^2, {x, -10, 10}, {y, -10, 10}];
ContourPlot[x^2 + y^2, {x, -10, 10}, {y, -10, 10}, ContourLines →False];
ContourPlot[x^2 + y^2, {x, -10, 10}, {y, -10, 10},ContourShading →False];
```

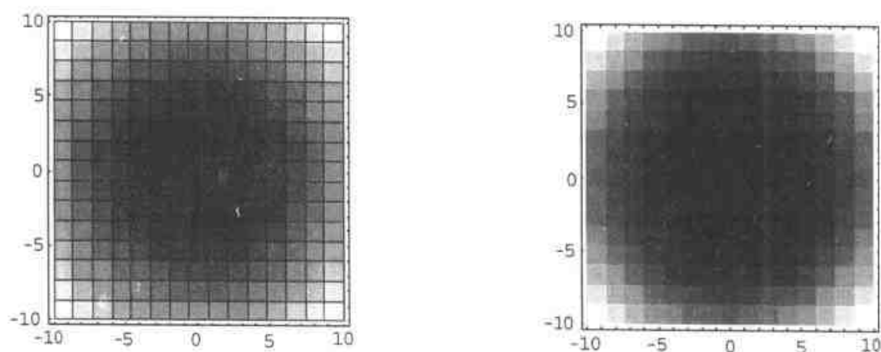
所谓密度图就是在点的规则数组中显示函数值.越淡的区域表示的值越高.

■ **DensityPlot[f[x, y], {x, xmin, xmax}, {y, ymin, ymax}]** 绘制函数 $f(x, y)$在由 xmin, xmax, ymin 与 ymax 确定的矩形上的密度图.**Mesh →False** 去掉分开加阴影区

域的矩形格线.



例 10
```
DensityPlot[x^2 + y^2, {x, -10, 10}, {y, -10, 10}];
DensityPlot[x^2 + y^2, {x, -10, 10}, {y, -10, 10}, Mesh -> False];
```



ListContourPlot、**ListDensityPlot** 与 **ContourPlot**、**DensityPlot** 是类似的，只是它们的参数为数的列表. 这两条命令适用于定义在整数坐标网格上的函数.

■ **ListContourPlot**[**array**] 生成二维数组的轮廓图.

■ **ListDensityPlot**[**array**] 生成二维数组的密度图.

其中 array = {{z_{11}, z_{12}, …}, {z_{21}, z_{22}, …}, …}表示在 x-y 平面上点的高度，它必须是嵌套的二维或更高维数组. z_{ij}是(j, i)点的 z 坐标. ListContourPlot 和 ListDensityPlot 的选项与 ContourPlot 和 DensityPlot 的选项是相同的，只是坐标轴的标签在默认情况下是从正整数 1 开始的. 选项 **MeshRange** 可以改变坐标轴的标签.

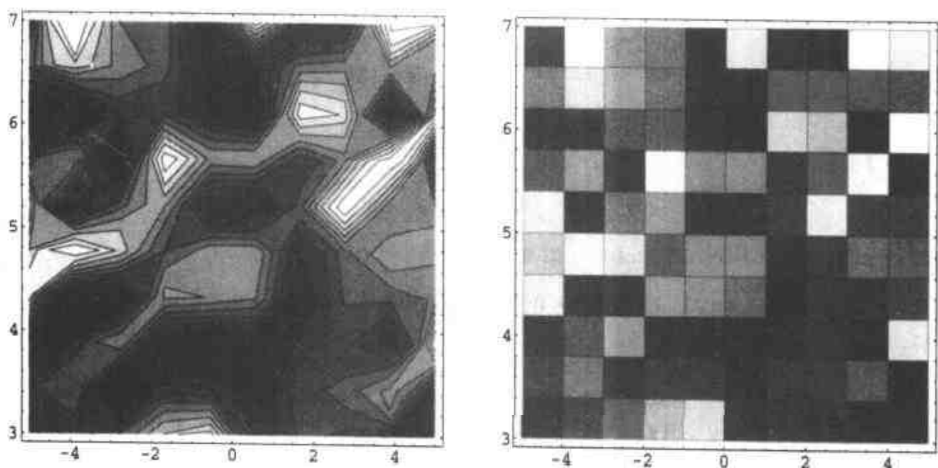
- **MeshRange** → {{**xmin, xmax**}, {**ymin, ymax**}} 分别为 x 轴与 y 轴加上从 xmin 到 xmax 以及从 ymin 到 ymax 的标签.

例 11
```
lst = Table[Random[],{x,1,10},{y,1,10}];     ←生成 10×10 的随机数组.
ListContourPlot[lst, MeshRange -> {{-5, 5}, {3, 7}}];
ListDensityPlot[lst, MeshRange -> {{-5, 5}, {3, 7}}];
```

ContourPlot3D 是 ContourPlot 在三维空间中的形式. ContourPlot3D 绘制函数 f 的层次曲面，即点集(x, y, z) 使得 $f(x, y, z) = k$ 成立.

■ **ContourPlot3D**[**f**[**x, y, z**], {**x, xmin, xmax**}, {**y, ymin, ymax**}, {**z, zmin, zmax**}] 在由 xmin, xmax, ymin, ymax, zmin 与 zmax 确定的盒子中绘出层次曲面 $f(x, y, z) = 0$ 的三维轮廓图.

ContourPlot3D 包含在软件包 Graphics`ContourPlot3D` 中，因此在使用前要上载这个软件包.

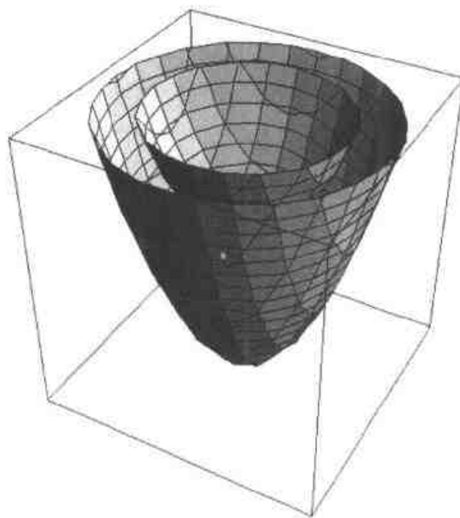


ContourPlot3D 的最常用选项为：

- **Contours** → {**k1, k2,** …} 绘制相应于 k1, k2, …的层次曲面. 默认值为 Contours→{0.}, 即只绘制一张层次曲面.
- **PlotPoints** → {**nx, ny**} 确定在 x 方向与 y 方向上分别要计算的点数. **PlotPoints** → **n** 等价于 **PlotPoints** → {n, n}. 默认值为 PlotPoints→{3, 5}.

例 12

```
<<Graphics`ContourPlot3D`
ContourPlot3D[z - x^2 - y^2, {x, -5, 5}, {y, -5, 5}, {z, 0, 10},
              Contours -> {0., 5.},
              BoxRatios -> {1, 1, 1},
              PlotPoints -> 5];
```



ListPlot3D 是 **ListPlot** 在三维空间中的形式.

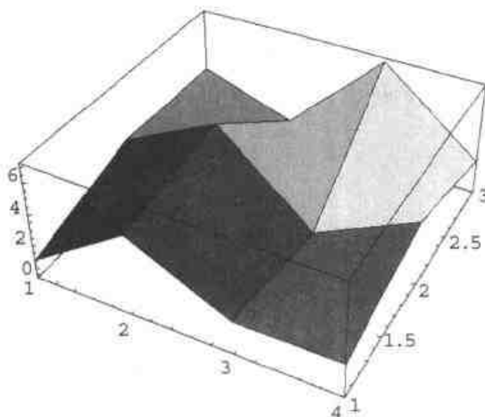
- **ListPlot3D**[**array**] 根据给定的高度数组, 生成三维空间中的曲面, 其中 array = {{z_{11}, z_{12}, …}, {z_{21}, z_{22}, …}, …}, z_{ij}表示点(j, i)的 z 坐标, 必须是至少2×2维的嵌套数组.
- **ListPlot3D**[**array, shades**] 生成曲面, 并且根据在数组 **shades** (GrayLevel, Hue, RGBColor) 中的描述加上阴影. 如果 array 的维数为 m×n, 那么 shades 必须具有维数 (m−1)×(n−1).

选项 **MeshRange** 控制 x 轴与 y 轴上的标签.

- **MeshRange** → {{**xmin, xmax**}, {**ymin, ymax**}} 分别为 x 轴与 y 轴加上从 xmin 到

xmax 以及从 ymin 到 ymax 的标签. 默认值为 **MeshRange →Automatic**, 它用从 1 开始的整数作为标签.

例 13
```
lst = {{1, 5, 2, 2}, {3, 6, 1, 4}, {3, 1, 7, 2}};
shades = {{GrayLevel[.4], GrayLevel[.5], GrayLevel[.6]},
         {GrayLevel[.7], GrayLevel[.8], GrayLevel[.9]}};
ListPlot3D[lst, shades];
```



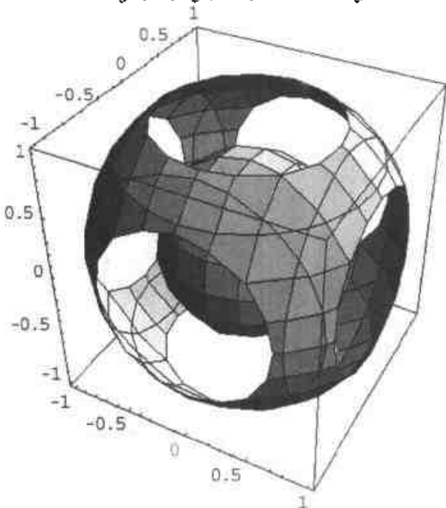
ContourPlot3D 的离散形式为 **ListContourPlot3D**.

- **ListContourPlot3D[array]** 绘制在 array 中数值的轮廓图, 其中 array 为表示 f 值的三维数组.

ListContourPlot3D 包含在软件包 **Graphics`ContourPlot3D`** 中, 因此必须在使用前上载这个软件包. ListContourPlot3D 的最常用选项为:

- **Contours →{k1, k2, …}** 绘制对应于函数值 k1, k2, …的轮廓. 默认值为 Contours →{0.}, 因此只绘制一个层次的曲面.
- **PlotPoints →{nx, ny}** 确定在 x 方向与 y 方向上分别要计算的点数. PlotPoints →n 等价于 PlotPoints→{n, n}. 默认值为 PlotPoints→{3, 5}.
- **Axes →True** 绘制坐标轴以及对应的标签. 默认值为 Axes→False.
- **MeshRange →{{xmin, xmax}, {ymin, ymax}, {zmin, zmax}}** 分别给 x 轴, y 轴, z 轴加上从 xmin 到 xmax, 从 ymin 到 ymax 以及从 zmin 到 zmax 的整数标签.

例 14 在这个例子中, 我们生成函数 $f(x, y, z) = x^2 + y^2 + z^2$ 的一组离散值, 并绘制对应于



$k=0.5$ 与 $k=1.5$ 的两个轮廓．所生成的曲面为球面，但大球被画在一个不能完整包含它的盒子中，这样就导致在图形中两个球都是可见的．

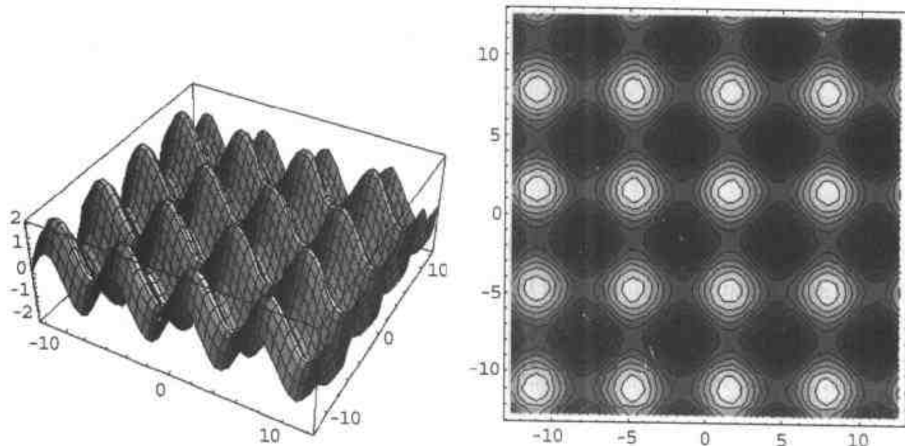
```
<<Graphics`ContourPlot3D`
lst = Table[x^2 + y^2 + z^2, {x, -1, 1, .25}, {y, -1, 1, .25}, {z, -1, 1, .25}];
ListContourPlot3D[lst, MeshRange→{{-1, 1}, {-1, 1}, {-1, 1}},
        Contours→{.5, 1.5}, Axes→True];
```

习题解答

5.12 给出 $f(x, y)=\sin x+\sin y$ 在正方形 $-4\pi\leqslant x,\ y\leqslant 4\pi$ 上的轮廓图，并与函数的三维图形作一对比．

解答

```
Plot3D[Sin[x] + Sin[y], {x, -4π, 4π}, {y, -4π, 4π}, PlotPoints→50];
ContourPlot[Sin[x] + Sin[y], {x, -4π, 4π}, {y, -4π, 4π}, PlotPoints→50];
```

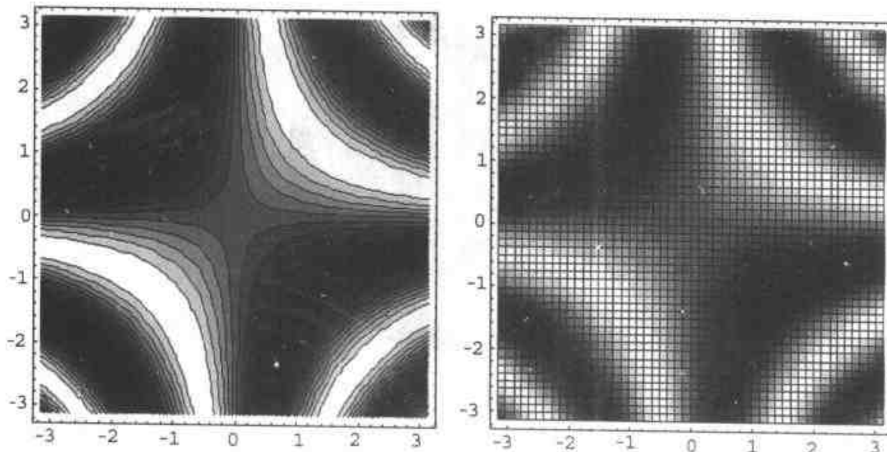


5.13 比较函数 $f(x, y)=\sin xy$ 在矩形 $-\pi\leqslant x,\ y\leqslant\pi$ 上的轮廓图与密度图．

解答

为了得到光滑的图形，我们使用 PlotPoints→50．

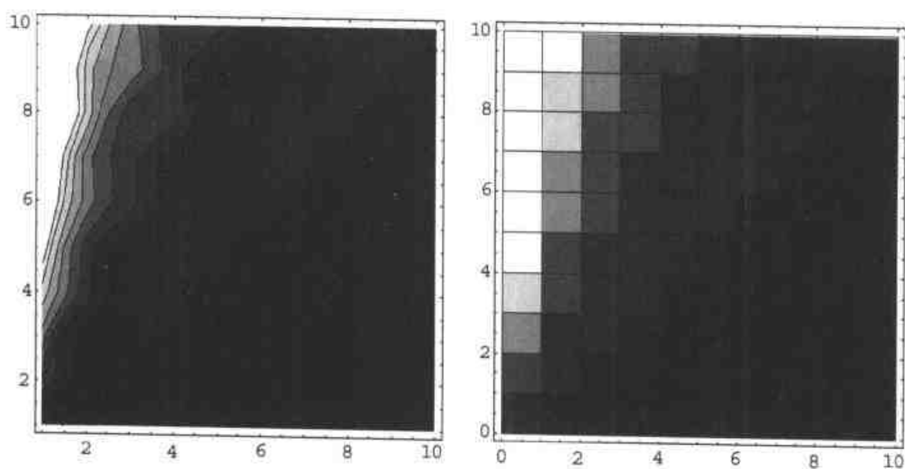
```
ContourPlot[Sin[x y], {x, -π, π}, {y, -π, π}, PlotPoints→50];
DensityPlot[Sin[x y], {x, -π, π}, {y, -π, π}, PlotPoints→50];
```



5.14 给出离散函数 Quotien[x, y]当 x 与 y 从 1 变到 10 时的轮廓图与密度图.

解 ☞

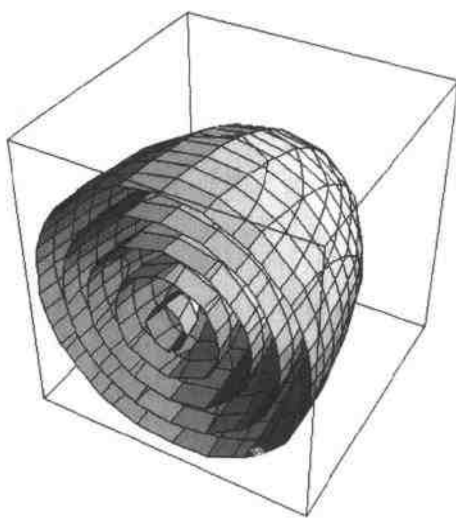
```
lst = Table[Quotient[x, y], {x, 1, 10}, {y, 1, 10}];
ListContourPlot[lst];
ListDensityPlot[lst];
```



5.15 令 $f(x, y, z) = 5x^2 + 2y^2 + z^2$. 画出当 $k = 1, 4, 9, 16, 25$ 时的层次曲面 $f(x, y, z) = k$. 只画出 $y \geqslant 0$ 的部分,从而所有的曲面都可见.

解 ☞

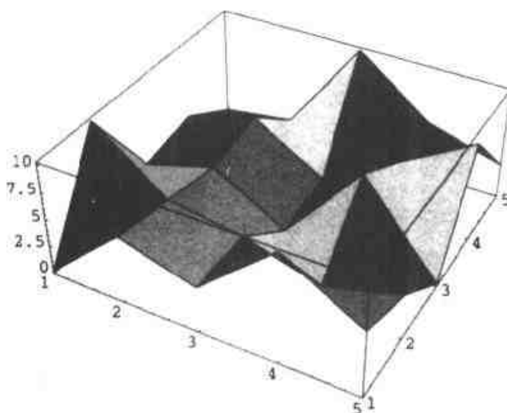
```
<<Graphics`ContourPlot3D`
ContourPlot3D[5x^2 + 2y^2 + z^2, {x, -5, 5}, {y, 0, 5}, {z, -5, 5},
              Contours -> {1, 4, 9, 16, 25},
              BoxRatios -> {1, 1, 1}, PlotPoints -> 5];
```



5.16 构造一个由介于 0 与 10 之间的随机数形成的 5×5 维数组,并作出这些值对应的列表图形.

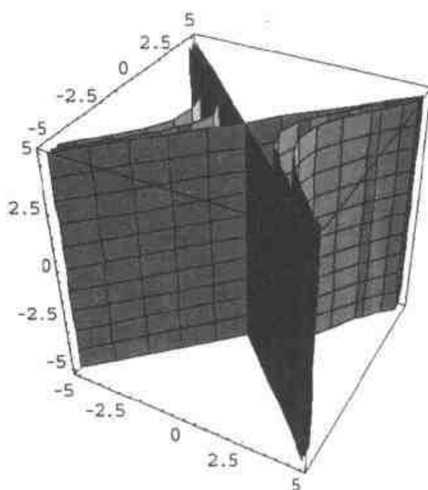
解 ☞

```
lst = Table[Random[Integer, {0, 10}], {x, 1, 5}, {y, 1, 5}];ListPlot3D[lst];
```



5.17 计算 $f(x, y, z) = x^2 - y^2$ 在介于 −5 到 5 之间整数点的值，并利用 ListContourPlot3D 画出柱面 $x^2 - y^2 = k, k = 0, 2, 5$ 的图形.

解 ☞



我们利用 x, y 与 z 的整数值构造列表 lst.

```
<<Graphics`ContourPlot3D`
lst = Table[x^2 - y^2, {z, -5, 5}, {y, -5, 5}, {x, -5, 5}];
ListContourPlot3D[lst, Contours->{0., 2., 5.}, Axes->True,
                  MeshRange->{{-5, 5}, {-5, 5}, {-5, 5}}];
```

5.3 特殊的三维图形

在软件包 **Graphics`Graphics3D`** 中包含大量实用的绘图命令. 当然，与所有的 Mathematica 软件包一样，在使用其中的命令之前，必须先上载这个软件包.

- **BarChart3D[array]** 利用 array 确定高度、从而构造三维的柱形图. 其中 array = {{z_{11}, z_{12}, …}, {z_{21}, z_{22}, …}, …}, z_{ij}表示柱(i, j)的高度.

任意三维图形选项都可以用在 BarChart3D 中. 另外，还可以使用下述选项：

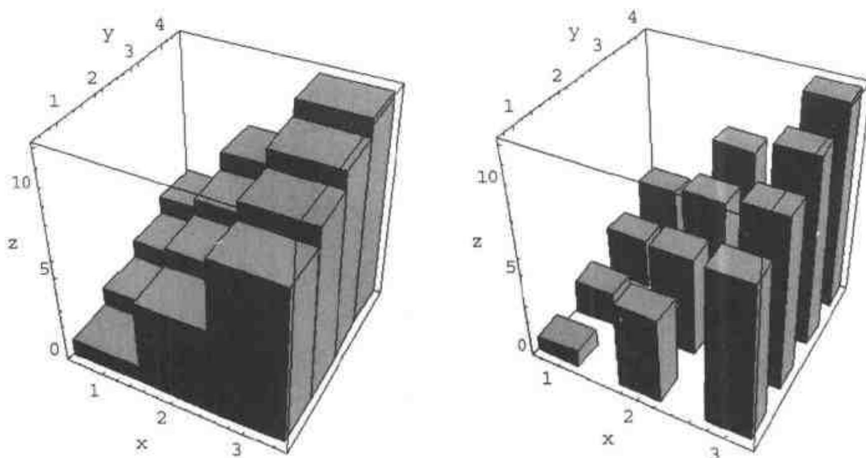
- **XSpacing** 与 **YSpacing** 确定在 x 方向与 y 方向上柱之间的距离. 它们的值必须介于

0 与 1 之间. 默认值为 **XSpacing →0** 与 **YSpacing →0**(即没有距离).

- **SolidBarEdges →False** 去掉柱之间的边界.
- **SolidBarEdgeStyle** 确定柱边界的绘制方式. 默认值为 SolideBarEdgeStyle → GrayLevel[0].
- **SolidBarStyle** 确定柱表面的样式. 默认值为 **SolidBarStyle →GrayLevel [0.5]**.

例 15

```
<<Graphics`Graphics3D`
lst = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}};
BarChart3D[lst, AxesLabel →{"x", "y", "z"}];
BarChart3D[lst, XSpacing →.5, YSpacing →.5, AxesLbel →{"x", "y", "z"}];
```



ScatterPlot3D 是 **ListPlot** 的三维形式, 后者在二维平面上绘制由离散点构成的图形.

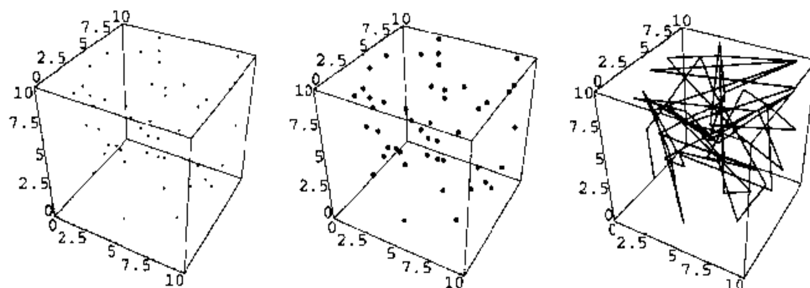
■ **ScatterPlot3D[lst]** 在三维盒子中绘制包含在 lst 中的点形成的图形. lst 必须是由子列表构成的列表, 每个子列表由三个数构成, 表示点的坐标.

ScatterPlot3D 接受在第四章讨论的 PlotStyle 选项. 另外, **PlotJoined →True** 把点连接起来. ScatterPlot3D 包含在 **Graphics`Graphics3D`** 软件包中.

在下面这个例子中, 我们生成 50 个随机点, 然后在三维空间中按不同的方法画出它的图形.

例 16

```
<<Graphics`Graphics3D`
lst = Table[Random[Integer, {1,10}], {50}, {3}];      ←生成 50 个随机点.
ScatterPlot3D[lst];
ScatterPlot3D[lst, PlotStyle →PointSize[.02]];
ScatterPlot3D[lst, PlotJoined →True];
```



ListSurfacePlot3D 从列表中指定的顶点构造多边形网格.

- **ListSurfacePlot3D[lst]** 由在 lst 中指定的顶点构造三维的多边形网格，其中 lst 形式为

  {{{x_{11}, y_{11}, z_{11}}, {x_{12}, y_{12}, z_{12}}, …}

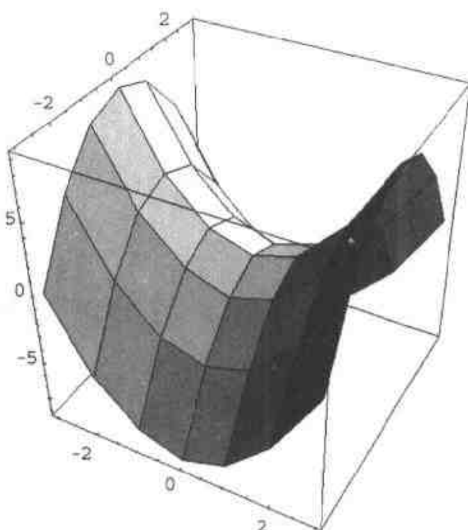
  {{x_{21}, y_{21}, z_{21}}, {x_{22}, y_{22}, z_{22}}, …}, …}.

例 17

```
<<Graphics`Graphics3D`
lst = Table[{x, y, x^2 - y^2}, {x, -3, 3}, {y, -3, 3}];
```

←创建在双曲抛物面 $z=x^2-y^2$ 上的 49 个点.

```
ListSurfacePlot3D[lst, Axes->True, BoxRatios->{1, 1, 1}];
```



旋转曲面就是由沿给定直线旋转一曲线得到的曲面. 命令 **SurfaceOfRevolution** 就可以构造这样的曲面. 根据选项的不同, 它具有各种形式. **SurfaceOfRevolution** 包含在软件包 **Graphics`SurfaceOfRevolution`** 中.

- **SurfaceOfRevolution[f[x], {x, xmin, xmax}]** 通过绕 z 轴旋转曲线 $z=f(x)$, xmin ≤x≤xmax 构造旋转面.
- **SurfaceOfRevolution[f[x], {x, xmin, xmax}, {θ, θ min, θ max}]** 通过绕 z 轴旋转曲线 $z=f(x)$, xmin ≤x≤xmax 构造部分旋转面, 即 θmin≤θ≤θmax.
- **SurfaceOfRevolution[{x[t], y[t], z[t]}, {t, tmin, tmax}]** 通过绕 z 轴旋转参数曲线 $x=x(t)$, $y=y(t)$, $z=z(t)$, tmin≤t≤tmax 构造旋转面.

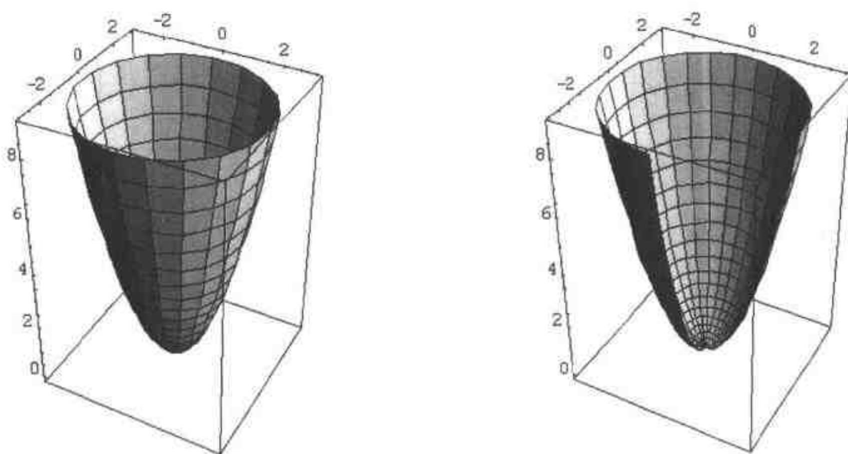
在下面的例子中绕 z 轴旋转曲面 $z=x^2$ 构造完整的或部分的旋转曲面.

例 18

```
<<Graphics`SurfaceOfRevolution`
SurfaceOfRevolution[x^2, {x, 0, 3}];
SurfaceOfrevolution[x^2, {x, 0, 3}, {θ, 0, 3π/2}];
```

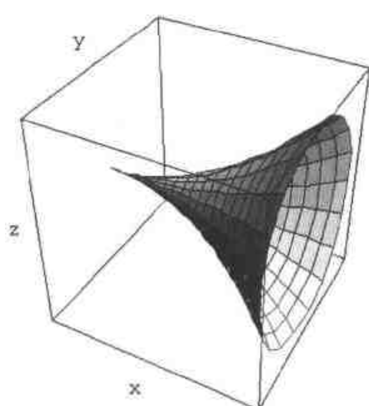
选项 **RevolutionAxis** 可以指定不同于 z 轴的其它旋转轴.

- **RevolutionAxis->{x, z}** 曲线沿 x-z 平面上原点与 (x, z) 点间连线旋转.
- **RevolutionAxis->{x, y, z}** 曲线沿空间中原点与 (x, y, z) 点间连线旋转.

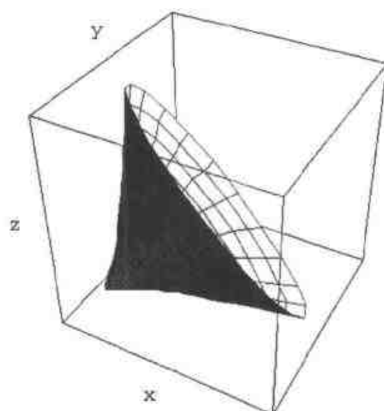


例 19

```
<<Graphics`SurfaceOfRevolution`
SurfaceOfRevolution[x^2, {x, 0, 3}, RevolutionAxis→{1, 0},
                    BoxRatios→{1, 1, 1},
                    Ticks→False,
                    AxesLabel→{"x", "y", "z"}];
SurfaceOfRevolution[x^2, {x, 0, 3}, RevolutionAxis→{1, 1, 1},
                    BoxRatios→{1, 1, 1},
                    Ticks→False,
                    AxesLabel→{"x", "y", "z"}];
```



曲线 $z=x^2$ 沿点 (0,0,0)
与点(1,0,0)的连线旋转.



曲线 $z=x^2$ 沿点 (0,0,0)
与点(1,1,1)的连线旋转.

习题解答

5.18 构造表示 $n = 5$ 时帕斯卡三角形的三维柱形图.

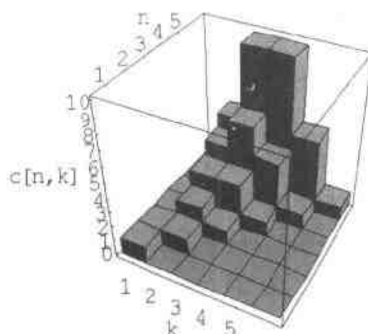
解 ☞

Pascal 三角形是二项式系数 $c(n, k)=\dfrac{n!}{k!\ (n-k)!}$ 的表示.

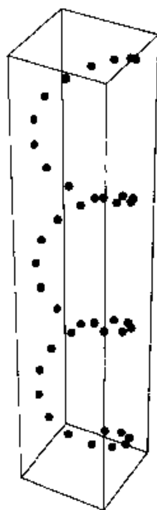
```
<<Graphics`Graphics3D`
c[n_, k_] = n!/(k! (n-k)!);          ←注意:如果 k > n, c[n,k]=0.
lst = Table[c[n, k], {k, 0, 5}, {n, 0, 5}];
```

		k					
		0	1	2	3	4	5
	0	1					
	1	1	1				
n	2	1	2	1			
	3	1	3	3	1		
	4	1	4	6	4	1	
	5	1	5	10	10	5	1

```
BarChart3D[lst, AxesLabel →{"k", "n", "c[n, k]"},
    Ticks →{Range[0, 5], Range[0, 5], Range[0, 10]}];
```



5.19 画出在螺旋线 $x = \sin 2t, y = \cos 2t, z = t$ 上的离散点图形,其中 t 从 0 变到 10,步长为 0.25.



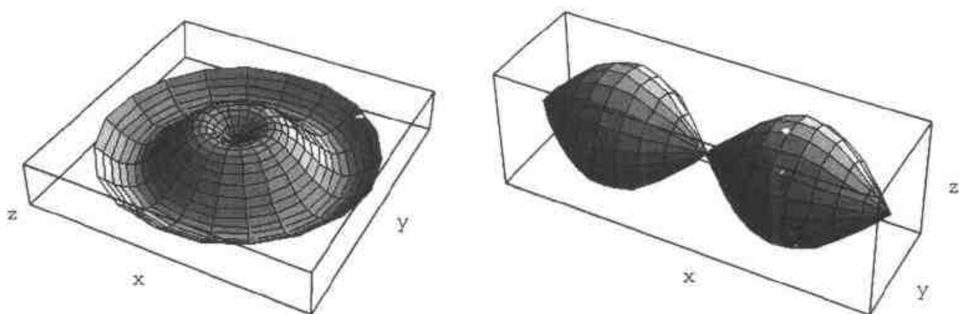
解 ☞

```
<<Graphics`Graphics3D`
lst = Table[{Sin[2t], Cos[2t], t}, {t, 0, 10, .25}];
ScatterPlot3D[lst, PlotStyle →PointSize[.05], Axes →
False];
```

5.20 利用曲线 $z = \sin x, 0 \leqslant x \leqslant 2\pi$ 构造旋转曲面,旋转轴分别为 (i) z 轴;(ii) x 轴.

解 ☞

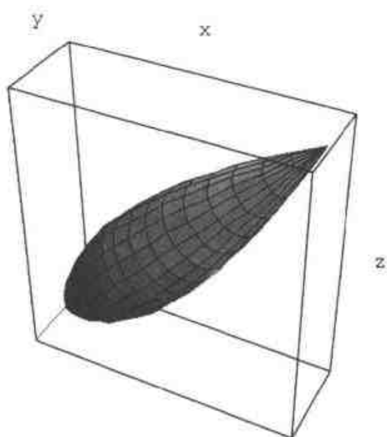
```
<<Graphics`SurfaceOfRevolution`
SurfaceOfRevolution[Sin[x], {x, 0, 2π}, Ticks →False,
                    AxesLabel →{"x", "y", "z"}];
SurfaceOfRevolution[Sin[x], {x, 0, 2π}, RevolutionAxis →{1, 0},
                    Ticks →False, AxesLabel →{"x", "y", "z"}];
```



5.21　画出沿直线 $z=x$ 旋转曲线 $z=x^2, 0\leqslant x\leqslant 1$ 所得旋转曲面的图形.

解

```
<<Graphics`SurfaceOfRevolution`
SurfaceOfRevolution[x^2, {x, 0, 1}, RevolutionAxis->{1, 1},
                    AxesLabel->{"x", "y", "z"}, Ticks->False];
```



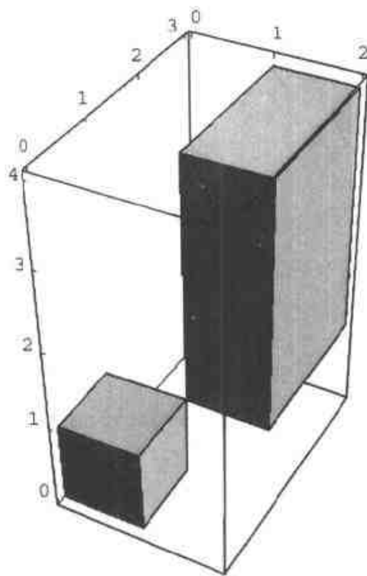
5.4　标准形状——三维图形基本单元

■ **Graphics3D[基本单元]** 或者 **Graphics3D[基本单元, 选项]** 创建三维的图形对象，可以用 Show 命令查看结果.

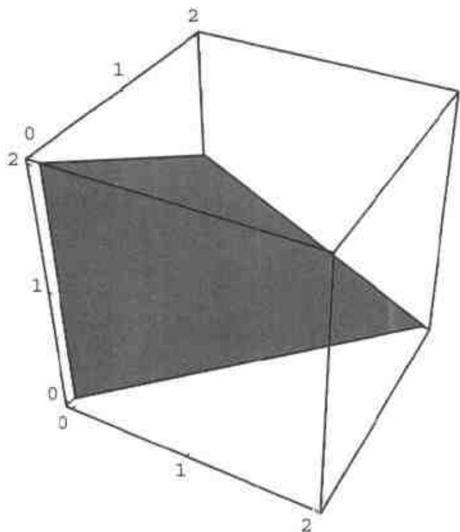
标准的基本单元有

- **Cuboid[{x, y, z}]** 为三维的图形基本单元，表示角点在(x, y, z)的单位长方体(立方体)，所有边线平行于坐标轴.
- **Cuboid[{x1, y1, z1}, {x2, y2, z2}]**表示长方体(平行六面体)，其对角点为(x1, y1, z1)和(x2, y2, z2).
- **Line[{x1, y1, z1}, {x2, y2, z2}, …]**画出连接(x1, y1, z1), (x2, y2, z2), … 的一系列线段.
- **Point[{x, y, z}]** 画出坐标为(x, y, z)的单个点.
- **Polygon[{x1, y1, z1}, {x2, y2, z2}, …]** 画出顶点坐标为(x1, y1, z1), (x2, y2, z2), … 的填充多边形.
- **Text[表达式, {x, y, z}]** 创建由中心位于(x, y, z)的表达式文本构成的图形基本单元.

例 20　Show[Graphics3D[{Cuboid[{0, 0, 0}], Cuboid[{1, 1, 1}, {2, 3, 4}]}],
Axes→True, Ticks→{{0, 1, 2},{0, 1, 2, 3},{0, 1, 2, 3, 4}}];



例 21　vertices = {{0, 0, 0}, {2, 2, 0}, {0, 2, 1}, {0, 0, 2}};
Show [Graphics3D[Polygon[vertices]], Axes→True,
Ticks→{{0, 1, 2}, {0, 1, 2}, {0, 1, 2}}];



除此而外,利用包含在 Graphics\`Shapes\`中的其它三维图形基本单元则可以方便地绘制类似于球面、锥面、柱面等标准形状.注意在使用前上载这个软件包.

- Cylinder[r, h, n] 使用 n 个多边形绘制一个圆柱,其半径为 r,半高为 h.
- Sphere[r, n, m] 使用 n(m−2)+2 个多边形绘制半径为 r 的球面.
- Cone[r, h, n] 使用 n 个多边形绘制底面半径为 r,半高为 h 的圆锥面.
- Torus[r1, r2, n, m] 使用 n×m 网格绘制半径为 r1 与 r2 的圆环面.
- MoebiusStrip[r1, r2, n] 使用 2n 个多边形绘制半径为 r1 和 r2 的 Möbius 带.
- Helix[r, h, m, n] 利用 n×m 网格绘制半径为 r,半高为 h,有 m 个来回的螺旋面.

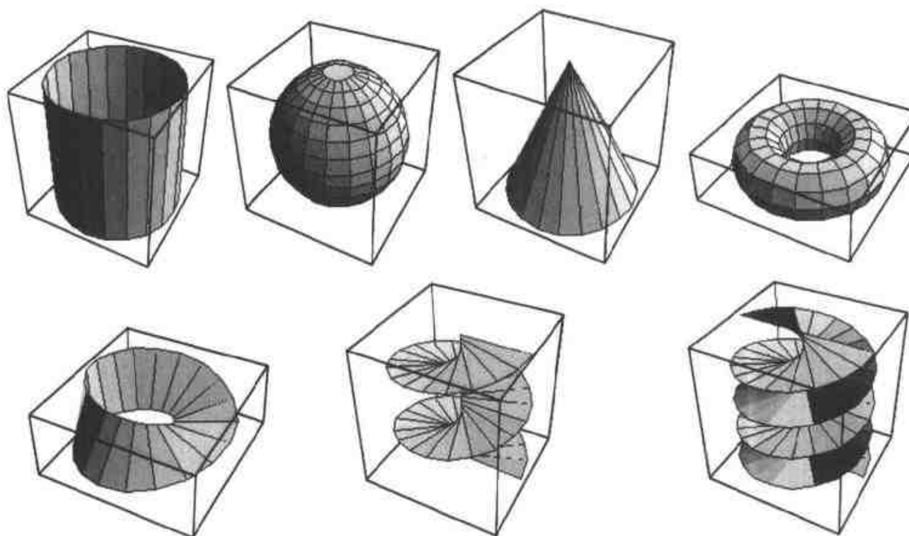
- **DoubleHelix[r, h, m, n]** 利用 n×m 网格绘制半径为 r，半高为 h，有 m 个来回的双螺旋面.

如果不给出参数，例如 Cone[ ]，Mathematica 就会使用其默认值.默认值分别为：

```
Cylinder[1, 1, 20]
Sphere[1, 20, 15]
Cone[1, 1, 20]
Torus[1, .5, 20, 10]
MoebiusStrip[1, .5, 20]
Helix[1, .5, 2, 20]
DoubleHelix[1, .5, 2, 20]
```

例 22

```
<<Graphics`Shapes`
Show[Graphics3D[Cylinder[]]];
Show[Graphics3D[Sphere[]]];
Show[Graphics3D[Cone[]]];
Show[Graphics3D[Torus[]]];
Show[Graphics3D[MoebiusStrip[]]];
Show[Graphics3D[Helix[]]];
Show[Graphics3D[DoubleHelix[]]];
```

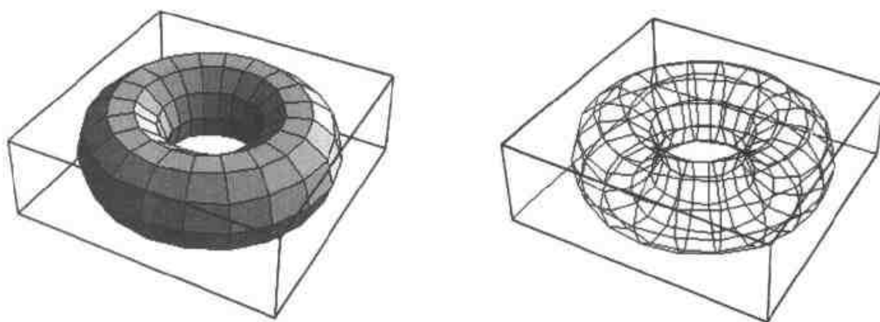


- **WireFrame[对象]** 把在构造对象时使用的全部多边形显示为透明的.可以作用到任何包含基本单位 Polygon，Line 与 Point 等的 Graphics3D 对象上.

例 23

```
<<Graphics`Shapes`
object = Graphics3D[Torus[]];
Show[object];
Show[WireFrame[object]];
```

在 Graphics`Shapes` 中有三条进行空间变换的命令：

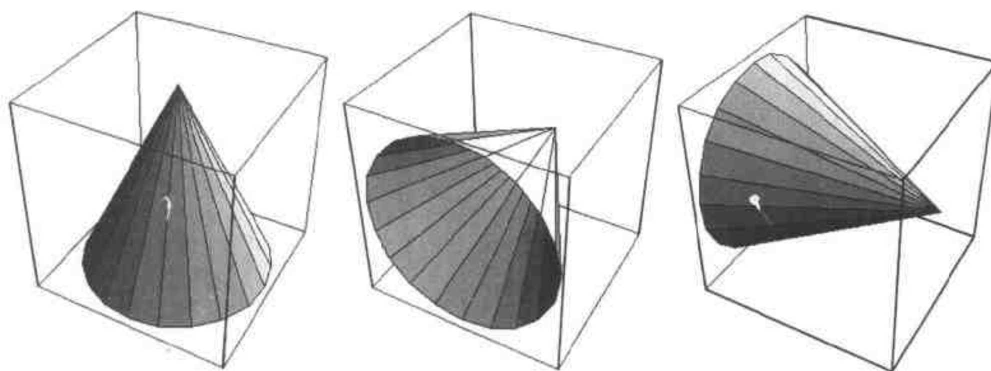


■ **RotateShape[对象, ϕ, θ, ψ]**　利用欧拉角[①] ϕ, θ, Ψ 旋转对象.

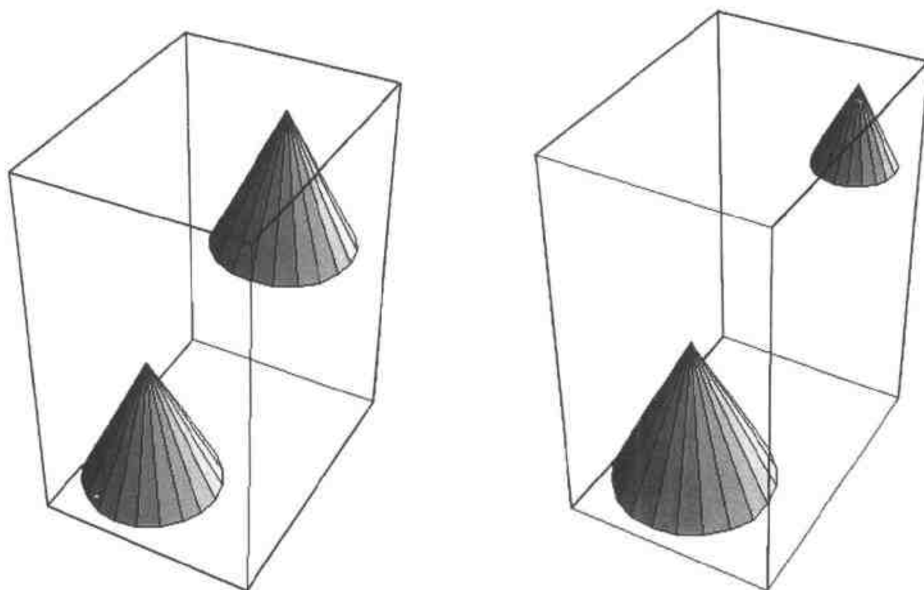
■ **TranslateShape[对象, {x, y, z}]** 把对象平移向量(x, y, z).

■ **AffineShape[对象, {xscale, yscale, zscale}]** 把 x, y, z 坐标分别进行因子为 xscale, yscale, zscale 的放缩变换.

例 24
```
<<Graphics`Shapes`
shape = Cone[];
object = Graphics3D[shape];
Show[object];
Show[RotateShape[object, 0, π/2, 0]];
Show[RotateShape[object, 0, π/2, π/2]];
Show[object, TranslateShape[object, {1, 2, 3}]];
shrunkenobject = AffineShape[object, {.5, .5, .5}];
Show[object, TranslateShape[shrunkenobject, {1, 2, 3}]];
```



① 欧拉角是描述在$\mathbb{R}^3$空间中按特定顺序进行三次旋转的变换方法。首先沿 z 轴旋转 ϕ 角，接着沿新的 y 轴旋转 θ 角，最后沿这些旋转所得的新 z 轴旋转 ψ 角.

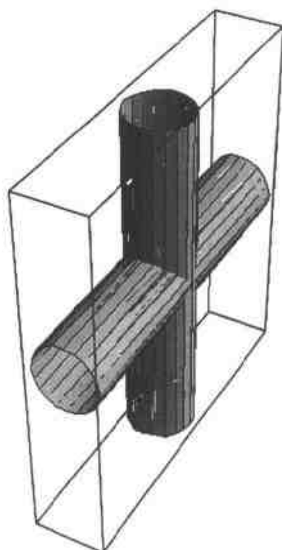


习题解答

5.22 绘制互相垂直的两个圆柱的图形.

解 ☞

```
<<Graphics`Shapes`
cyl1 = Graphics3D[Cylinder[1, 5, 20]];
cyl2 = Graphics3D[RotateShape[Cylinder[1, 5, 20], 0,π/2, 0]];
Show[cyl1, cyl2];
```

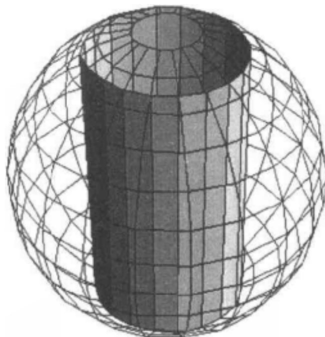


5.23 构造一个位于半径为1的球面内部的圆柱.

解 ☞

由于球面的半径为1,所以使用默认参数.为了使圆柱包含在球面内部,r(半径)与h(高度)必须满足$r^2+h^2=1$.这里选择$r=1/2, h=\sqrt{3}/2$.为了使圆柱可见,我们把球面画成线框样式.

```
<<Graphics`Shapes`
sph = WireFrame[Graphics3D[Sphere[]]];
cyl = Graphics3D[Cylinder[1/2, √3/2, 20]];
Show[sph, cyl, Boxed→False];
```

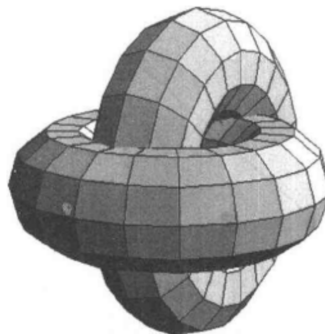


5.24 绘制两个嵌套的圆环面.

解 ☞

我们使用两个默认尺寸的圆环面(r1 = 1, r2 = 0.5),第二个圆环面必须旋转 90°,并且平移 0.5 个单位,以使得它们互相嵌套,但不相交.

```
<<Graphics`Shapes`
tor1 = Graphics3D[Torus[]];
tor2 = Graphics3D[TranslateShape[
          RotateShape[Torus[], 0,π/2,π/2], {0, .5, 0}]];
Show[tor1, tor2, ViewPoint→{1.75, -2.8, 0.75}, Boxed→False];
```



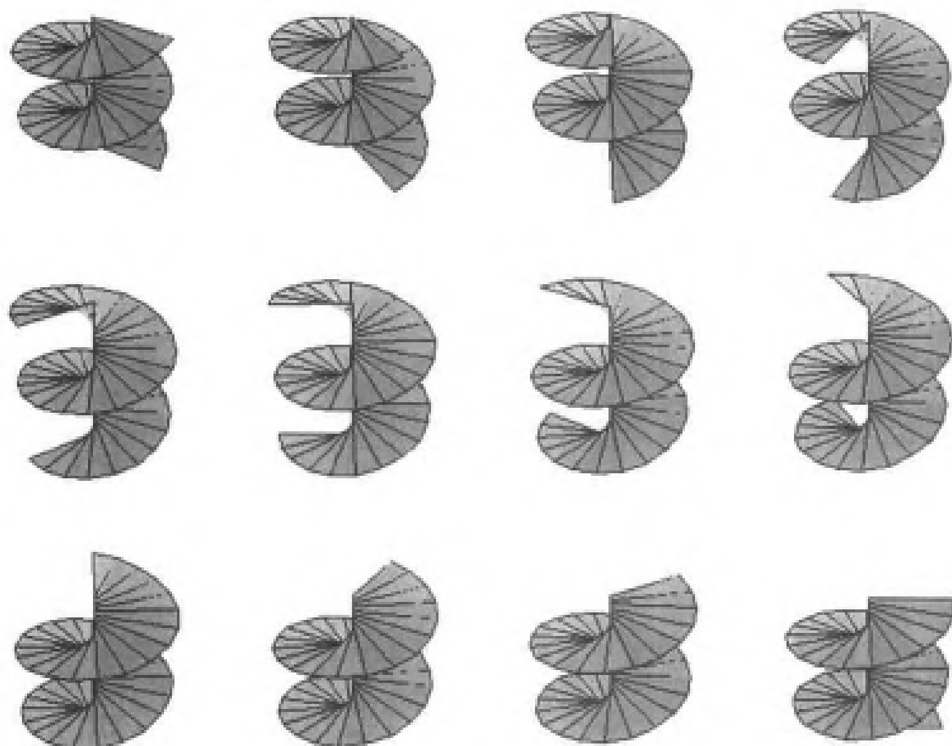
5.25 构造一个动画,演示螺旋线绕 z 轴盘旋的过程.

解 ☞

我们使用缺省尺寸的螺旋线 helix[]. 当欧拉角 ϕ 从 0 变到 2π 时,螺旋线完成一次旋转. 下面使用 12 帧图像进行动画演示.

```
<<Graphics`Shapes`
helx = Helix[];
rotatedhelix = RotateShape[helx, k, 0, 0];
Do[Show[Graphics3D[rotatedhelix], Boxed→False, {k, 0, 11π/6,π/6}]
```

选择这些图像的单元括号,然后进入 **Cell ⇒ Animate Selected Graphics** 就可以实现动画演示了.



第六章　求 解 方 程

6.1　求解代数方程

一般代数方程的解可以用 **Solve** 命令求出. 这个命令非常容易使用,但必须注意一点,在方程左右两边之间要使用双等号 == .(在前面讲过,双等号表示逻辑等式:lhs == rhs 当且仅当 lhs 与 rhs 有相同值时,值为 True,否则值为 False.)

由 Solve 确定的根是用下述形式的列表表示的:

{{x→x1}, {x→x2}, …}

记号 x→x1 表示 x 的解为 x1,但 x 并没有被 x1 取代. 如果方程有一个根重数 $m>1$,那么就重复出现 m 次.

■　**Sovle[方程, 变量]** 尝试求解关于变量的方程.

对于方程组,方程的形式为{方程 1, 方程 2, …},变量既可以是单个变量,也可以是几个变量构成的列表. 另外,方程也可以是用 && (逻辑与)隔开的单个方程表示. 如果只出现了一个变量,变量参数可以不出现.

例 1　在这个例子中,由于只有一个变量,因此不需要给出变量参数.

Solve[7x + 3 == 3x + 8]

$\left\{\left\{x\to\frac{5}{4}\right\}\right\}$

如果要求解的方程中包含两个或更多的变量,那么我们就必须指定要相应于哪个变量求解.

例 2　**Solve[a y + b == c x + d]**

> 注意 a 与 y 以及 c 与 x 之间的空白,此处的空白相当重要,可以用 * 代替空白.

Solve::svars:

Equations may not give solutions for all "solve" variables.

(不能给出方程相应于所有求解变量的解.)

{{b→d + c x − a y}}

我们必须指定要求解的变量.

Solve[a y + b == c x + d, x]

$\left\{\left\{x\to-\frac{-b+d-a\,y}{c}\right\}\right\}$

Solve[a y + b == c x + d, y]

$\left\{\left\{y\to-\frac{b-d-c\,x}{a}\right\}\right\}$

Solve[a y + b == c x + d, b]

{{b→d + c x − a y}}

Solve[a y + b == c x + d, d]

{{d→b − c x + a y}}

例 3　下面这个例子演示了如何求解简单方程组:$\begin{cases}2x+3y=7\\3x+4y=10\end{cases}$的方法.

```
Solve[{2x + 3y == 7, 3x + 4y == 10}, {x, y}]或者
Solve[2x + 3y == 7 && 3x + 4y == 10}, {x, y}]
{{x→2, y→1}}
```

这里没有必要指定{x, y}，因为我们有的是两个变量的两个方程

如果未知数的个数超过方程的个数，那么就必须指定所希望求解的变量. 否则就会得到 Mathematica 的默认处理结果.

例 4 `Solve[{x + 2y + z == 5, 2x + y + 3z == 7}, {y, z}]`

$\left\{\left\{y\to\frac{8-x}{5}, z\to-\frac{3}{5}(-3+x)\right\}\right\}$

当然，Solve 并不是只能求解线性方程.

例 5 `Solve[a x^2 + b x + c == 0, x]`

$\left\{\left\{x\to\frac{-b-\sqrt{b^2-4ac}}{2a}\right\}, \left\{x\to\frac{-b+\sqrt{b^2-4ac}}{2a}\right\}\right\}$

注意　Mathematica 用任意的 a, b, c 表达出方程的一般解.

例 6 `Solve[x^3 + y^2 == 5 && x + y == 3]`

$\{\{y\to2, x\to1\}, \{y\to4-\sqrt{5}, x\to-1+\sqrt{5}\}, \{y\to4+\sqrt{5}, x\to-1-\sqrt{5}\}\}$

由于 Mathematica 把方程的解表示为嵌套列表，因此不能直接把它作为其它数学结构的输入. 但是有两种方法可以调用其中的值，而不必采用照抄或粘贴的方法.

(a) 如果希望利用由 Solve 得到的解计算表达式的值，可以利用取代运算符/.，这样 Mathematica 就会自动代入相应的值.

(b) 由于解就是列表，因此可以用 Part 或[[ ]]从列表中"提取"解.

在下面两个例子中演示这些方法的使用.

例 7　假设希望求解方程组 $\begin{cases}x^2+y=5,\\x+y=3,\end{cases}$ 并计算表达式 $\sqrt{x^2+y^2}$ 在这些解上的值. 为此下面首先利用 Solve 命令计算方程组的解，为了简单起见，把解记为 solutions.

```
solutions = Solve[{x^2 + y == 5, x + y == 3}, {x, y}]
{{y→1, x→2}, {y→4, x→-1}}
```

$\sqrt{x^2+y^2}$`/.solutions`

$\{\sqrt{5}, \sqrt{17}\}$　　←Mathematica 生成一个列表，包含表达式所有的值.

例 8　假设要计算下述方程所有根的平方和：

$$x^6-21x^5+175x^4-735x^3+1624x^2-1764x+720=0.$$

为此，先利用 Solve 命令求解出方程的所有根.

```
solutions = Solve[x^6 - 21x^5 + 175 x^4 - 735 x^3 + 1624 x^2 - 1764 x + 720 == 0]
{{x→1}, {x→2}, {x→3}, {x→4}, {x→5}, {x→6}}
```

仔细查看 solutions，可知它是包含子列表的列表. 先看其第一部分.

```
solutions[[1]]
{x→1}
```

由于这个列表只有一部分，我们可以提取它的内容.

```
solutions[[1, 1]]
```

x→1

为了得到这个表达式的第二部分(箭头后面的数),我们进一步地提取:

solutions[[1, 1, 2]]

1

为了理解为什么会这样,我们深入地查看一下 solutions[[1, 1]]的结构.

FullForm[solutions[[1, 1, 2]]]

Rule[x, 1]

x→1 就等价于 Rule[x,1],从而可以用 solutions[[1, 1, 2]]提取出它的第二个参数.类似地,其它解可以用 solutions[[2, 1, 2]],solutions[[3, 1, 2]]等表达式提取出来.为了得到它们的平方和,输入

Sum[solutions[[k, 1, 2]]^2, {k,1,6}]或者 $\sum_{k=1}^{6}$ **solutions[[k,1,2]]**^{2}

91

Solve 的设计方法,决定了它可以求解代数方程,但有时候它也可以求出超越方程的有限解.这时会得到一条警告消息,说明它并没有求出所有的解.

例 9　**Solve[Sin[x] == 1/2, x]**

Solve::ifun:

Inverse functions are being used by Solve, so some solutions may not be found.

(Solve 使用了反函数,因此有可能有些解没有求出.)

$\left\{\left\{x\to\frac{\pi}{6}\right\}\right\}$

如果要求解的方程是矛盾的,那么 Mathematica 就会返回空白列表.

例 10　**Solve[{2x + 3y == 5,4x + 6y ==11}]**

{ }

如果方程的解中包含复数,那么就用 −1 的幂次表示.然而如果希望得到更符合惯例的表示,那么可以在 Solve 命令的右边插入

/.(a_→b_) :>(a→ComplexExpand[b])

这里:>指的是 RuleDelayed.

如果只希望得到实根,那么可以上载 **Miscellaneous`RealOnly`** 软件包.这样就会给所有的复根标上`Nonreal`.

例 11　**Solve[x^3 ==1]**

{{x→1}, {x→−(−1)$^{1/3}$}, {x→(−1)$^{2/3}$}}

Solve[x^3 ==1]/.(a_→b_) :>(a→ComplexExpand[b])

$\left\{\{x\to 1\},\left\{x\to-\frac{1}{2}-\frac{i\sqrt{3}}{2}\right\},\left\{x\to-\frac{1}{2}+\frac{i\sqrt{3}}{2}\right\}\right\}$

<<Miscellaneous`RealOnly`

Solve[x^3 == 1]

Nonreal::warning: Nonreal number encountered. (遇到了非实数.)

{{x→1}, {x→Nonreal}, {x→Nonreal}}

对于所有的代数方程而言,即使理论上解是存在的,但 Mathematica 也有可能无法求出它的解来.如果 Mathematica 无法求解一个方程,它就用符号形式表示解.对于大多数情形,这种

形式解是没有任何用处的,因此给出数值解就更恰当了.数值近似解可以用 **NSolve** 命令求出.

■ **NSolve[方程, 变量]** 求解方程关于变量的数值解.

■ **NSolve[方程, 变量, n]** 求解方程关于变量的数值解,达到 n 位精确度.

与 Solve 一样,如果没有歧义的话,可以不给出变量列表参数.

例 12
```
equation = x^4 - 16x^3 + 61x^2 - 22x - 12 == 0;
Solve[equation]
```
为了避免重复输入,用 equation 表示要求解的方程.

$\{\{x\to 3-\sqrt{5}\},\{x\to 3+\sqrt{5}\},\{x\to 5-2\sqrt{7}\},\{x\to 5+2\sqrt{7}\}\}$

```
NSolve[equation]
{{x→ -0.291503}, {x→0.763932}, {x→5.23607}, {x→10.2915}}
NSolve[equation, 25]
{{x→ -0.2915026221291811810032315},
{x→0.7639320225002103035908263},
{x→5.236067977499789696409174},
{x→10.29150262212918118100323}}
```

有时候,如果所要求的有效数字很多,那么 NSolve 可能无法给出解.这是因为 Mathematica 在内部计算时采用 16 位有效数字.对于这种情况,**N[Solve[方程, 变量], n]**就完全行得通,其中 n 为所要求的有效数字位数.

例 13 NSolve[$\sqrt[3]{x}+\sqrt{x}$ + x == 10, x, 20]

{{x→5.79617}} ←由于精度缺乏,最后只给出了 6 位有效数字.

N[Solve[$\sqrt[3]{x}+\sqrt{x}$ + x == 10, x], 20]

{{x→5.7961701257629574569}}

所谓增根,就是一个并不是方程真正根的数,但却在求解过程中得到了它.当求解根式方程时,就会遇到增根.例如,在求解$\sqrt{x}=-3$时,它虽然没有实根,但采取平方求解过程就会得到增根 $x=9$.

- **VerifySolutions** 选项决定是否要 Mathematica 判断所得解是否为增根.默认值 **VerifySolutions→True** 就会从解表中删去增根.如果需要这样的增根,那么就可以使用 **VerifySolutions→False** 选项.

例 14 Solve[x + $\sqrt{x}$ == 5]

$\left\{\left\{x\to \frac{1}{2}(11-\sqrt{21})\right\}\right\}$

Solve[x + $\sqrt{x}$ == 5, VerifySolutions→False]

$\left\{\left\{x\to \frac{1}{2}(11-\sqrt{21})\right\},\left\{x\to \frac{1}{2}(11+\sqrt{21})\right\}\right\}$

对于关于 x 的方程 $ax=b$, Solve 会给出解 $x=b/a$.然而如果 $a=b=0$,那么每个数 x 都是它的解.**Reduce** 命令可以用来描述所有可能的解.

■ **Reduce[方程, 变量]** 简化方程,并尝试求解**变量**.如果**方程**为恒等式,**Reduce** 就返回值 True.如果**方程**为矛盾的,Reduce 就返回值 False.

在解的描述中,Reduce 使用符号 &&(逻辑与)和||(逻辑或).&& 优先于||.

例 15 Solve[a x == b, x]

```
{{x→b/a}}
Reduce[a x == b, x]
a == 0 && b == 0||x == b/a && a≠0
```
←$a=b=0$ 或 $a\neq 0$, $x=b/a$.

```
Reduce[x^2 - 9 == (x + 3)(x - 3), x]
True
Reduce[x^2 - 10 == (x + 3)(x - 3), x]
False
```

习题解答

6.1　求出过 (2, 5) 与 (7, 9) 两点的直线方程.

解 ☞

直线的一般方程为 $y=ax+b$,代入两点的坐标,就得到方程 $2a+b=5$ 与 $7a+b=9$.

```
Solve[2a + b == 5 && 7a + b == 9]
{{a→4/5, b→17/5}}
```

因此直线的方程为 $y=\frac{4}{5}x+\frac{17}{5}$.

6.2　求出过 (1, 4), (2, 7) 与 (4, 11) 三点的圆的方程.

解 ☞

圆的一般方程为 $x^2+y^2+ax+by+c=0$. 代入给定点的坐标,就得到方程 $17+a+4b+c=0, 53+2a+7b+c=0, 137+4a+11b+c=0$.

```
Solve[{17 + a + 4b + c == 0, 53 + 2a + 7b + c == 0,
      137 + 4a + 11b + c == 0}]
{{a→ - 54, b→6, c→13}}
```

因此圆的方程为 $x^2+y^2-54x+6y+13=0$.

6.3　解方程 $x^5+x^4+x^3+x^2+x+2=0$.

解 ☞

```
Solve[x^5 + x^4 + x^3 + x^2 + x + 2 == 0]
{{x→Root[2 + #1 + #1^2 + #1^3 + #1^4 + #1^5&, 1]},
 {x→Root[2 + #1 + #1^2 + #1^3 + #1^4 + #1^5&, 2]},
 {x→Root[2 + #1 + #1^2 + #1^3 + #1^4 + #1^5&, 3]},
 {x→Root[2 + #1 + #1^2 + #1^3 + #1^4 + #1^5&, 4]},
 {x→Root[2 + #1 + #1^2 + #1^3 + #1^4 + #1^5&, 5]}}
```

由于 Mathematica 无法精确地求解这个方程,因此它给出符号解.然而,我们可以给出它的近似解.

```
NSolve[x^5 + x^4 + x^3 + x^2 + x + 2 == 0]
{{x→ - 1.21486}, {x→ - 0.522092 - 1.06118 i},
 {x→ - 0.522092 + 1.06118 i}, {x→0.629523 - 0.883585 i},
 {x→0.629523 + 0.883585 i}}
```

6.4 求解关于 w, x, y 的下述方程组,然后再确定当 $z=1, z=2, z=3$ 时的解.

$$\begin{cases} w + x + y + z = 3, \\ 2w + 3x + 4y + 5z = 10, \\ w - x - y + z = 4. \end{cases}$$

解 ☞

```
equations = {w + x + y + z == 3, 2w + 3x + 4y + 5z == 10, w - x + y - z == 4}
solutions = Solve[equations, {w, x, y}]
```

$\left\{\left\{w \to \frac{1}{4}(5+4z), x \to \frac{1}{2}(-1-2z), y \to \frac{1}{4}(9-4z)\right\}\right\}$

```
solutions/. z →1
```

$\left\{\left\{w \to \frac{9}{4}, x \to -\frac{3}{2}, y \to \frac{5}{4}\right\}\right\}$

```
solutions /. z →2
```

$\left\{\left\{w \to \frac{13}{4}, x \to -\frac{5}{2}, y \to \frac{1}{4}\right\}\right\}$

```
solutions /. z →3
```

$\left\{\left\{w \to \frac{17}{4}, x \to -\frac{7}{2}, y \to \frac{3}{4}\right\}\right\}$

6.5 求出一个数,其自身与其平方、立方的总和等于 30,要求所得结果具有 20 位有效数字.

解 ☞

```
NSolve[x + x^2 + x^3 == 30, x, 20]
{{x→ - 1.8557621138713175532 - 2.7604410593413850003 i},
 {x→ - 1.8557621138713175532 + 2.7604410593413850003 i},
 {x→2.7115242277426351064}}
```

6.6 对于三角函数方程 $2\sin^2 x+1=3\sin x$,分别关于 $\sin x$ 与 x 进行求解.

解 ☞

为了关于 $\sin x$ 求解,可以使用

```
Solve[2 Sin[x]^2 + 1 == 3Sin[x]]
```

$\left\{\left\{\text{Sin}[x] \to \frac{1}{2}\right\}, \{\text{Sin}[x] \to 1\}\right\}$

如果要对于 x 求解,那么只会得到主值解(利用反函数).

```
Solve[2 Sin[x]^2 + 1 == 3 Sin[x], x]
Solve::ifun:
Inverse functions are being used by Solve, so some solutions may
not be found.
```

(Solve 使用了反函数,因此有可能有些解没有求出.)

$\left\{\left\{x \to \frac{\pi}{6}\right\}, \left\{x \to \frac{\pi}{2}\right\}\right\}$

6.7 求解关于 x 的方程:$e^{2x}+e^x=3$.

解 ☞

```
Solve[Exp[2x] + Exp[x] == 3, x]
Solve::ifun:
Inverse functions are being used by Solve, so some solutions may
```

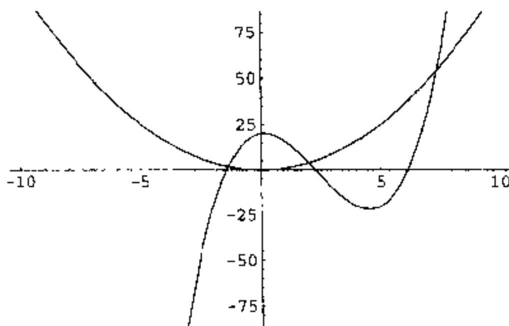
not be found.(Solve 使用了反函数,因此有可能有些解没有求出.)

$\left\{\left\{x \to \text{Log}\left[\frac{1}{2}(-1+\sqrt{13})\right]\right\},\left\{x \to i\pi + \text{Log}\left[\frac{1}{2}(1+\sqrt{13}\right]\right\}\right\}$

6.8 在同一坐标系中画出函数 $f(x)=x^3-7x^2+2x+20$ 与 $g(x)=x^2$ 的图形,并分别准确和近似地求出它们的交点.

解 ☞

```
f[x_] = x^3 - 7x^2 + 2x + 20;
g[x_] = x^2;
Plot[{f[x], g[x]}, {x, -10, 10}];
```




```
xvalues = Solve[f[x] == g[x], x];
{x, f[x]}/.xvalues //Simplify
```

$\{\{2,4\},\{3-\sqrt{19},28-6\sqrt{19}\},\{3+\sqrt{19},28+6\sqrt{19}\}\}$

```
% //N
```

{{2.,4.}, {-1.3589, 1.84661}, {7.3589, 54.1534}}

6.9 在代数中有一个定理,说的是如果 $p(x)=a_nx^n+a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x+a_0$,那么方程 $p(x)=0$ 的所有根的和为 $-\frac{a_{n-1}}{a_n}$,乘积为 $(-1)^n\frac{a_0}{a_n}$.对于下述方程验证这个定理:

$$20x^7+32x^6-221x^5-118x^4+725x^3-18x^2-726x+252=0.$$

解 ☞

```
solution = Solve[20x^7 + 32 x^6 - 221 x^5 - 118 x^4 + 725 x^3 - 18x^2 - 726 x + 252 == 0]
```

$\left\{\left\{x\to-\frac{7}{2}\right\},\left\{x\to\frac{2}{5}\right\},\left\{x\to\frac{3}{2}\right\},\{x\to-\sqrt{2}\},\{x\to\sqrt{2}\},\{x\to-\sqrt{3}\},\{x\to\sqrt{3}\}\right\}$

`Sum[solution[[k, 1, 2]], {k, 1, 7}]` 或者 $\sum_{k=1}^{7}$ `solution[[k,1,2]]`

$-\frac{8}{5}$　　　　$\leftarrow n=7; -\frac{a_{n-1}}{a_n}=-\frac{32}{20}=-\frac{8}{5}$.

`Product[solution[[k, 1, 2]], {k, 1, 7}]` 或者 $\prod_{k=1}^{7}$ `solution[[k,1,2]]`

$-\frac{63}{5}$　　　　$\leftarrow n=7; (-1)^n\frac{a_0}{a_n}=(-1)^7\frac{252}{20}=-\frac{63}{5}$.

6.10 求出关于 x 的方程 $ax-b=cx+d$ 的所有可能解.

解

```
Solve[a x + b == c x + d, x]
```

$\left\{\left\{x \to -\frac{b-d}{a-c}\right\}\right\}$

这个解需要前提 $a \neq c$. 可以用 Reduce 得到更一般的解.

```
Reduce[a x + b == c x + d, x]
```

$a == c \ \&\& \ b == d \ || \ x == \frac{-b+d}{a-c} \ \&\& \ a - c \neq 0$

6.2 求解超越方程

所谓超越方程就是那些不属于代数方程的方程. 虽然在有限的情况中, 可以用 Solve 与 NSolve 处理简单的三角函数方程或者指数方程, 但这些命令本来就不是设计用来求解包含复杂超越函数的方程的. Mathematica 命令 **FindRoot** 是处理这些方程的更佳工具.

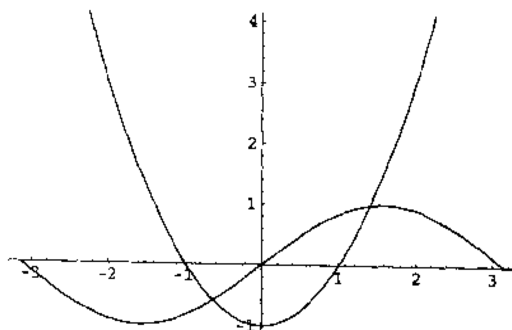
FindRoot 利用迭代方法计算方程的根, 因此必须指定起始值, 有时也称之为初始估计值. 为了得到最好的结果, 初始估计值应当尽可能地靠近所期望的根.

- **FindRoot[*lhs* == *rhs*, {x, x0}]** 利用初始值为 x0 的牛顿方法求解方程 *lhs* == *rhs*.
- **FindRoot[*lhs* == *rhs*, {x, x0, x1}]** 利用初始值为 x0 和 x1 的(修正)截线法求解方程 *lhs* == *rhs*.
- **FindRoot[*lhs* == *rhs*, {x, x0, xmin, xmax}]** 尽力求解方程, 但是如果迭代超出了区间[xmin, xmax]就会停止.

如果在 *lhs* == *rhs* 处指定的是一个函数, 那么 FindRoot 就会计算函数的零点. 所谓函数 f 的零点, 就是满足 $f(x)=0$ 的数 x.

例 16 方程 $\sin x = x^2 - 1$ 有两个根. 从左右两个函数的图形可知, 它们的交点靠近 $x=-1$ 和 $x=1$.

```
Plot[{Sin[x], x^2 - 1}, {x, -π, π}];
```




```
FindRoot[Sin[x] == x^2 - 1, {x, -1}]
    {x→ -0.636733}
FindRoot[Sin[x] == x^2 - 1, {x,1}]
{x→1.40962}
```

例 17 方程 $x + \sin x = 0$ 只有惟一解 $x=0$. 下面看看如果初始估计值离解太远的话, 会发生什么事情.

```
FindRoot[x + Sin[x] == 0, {x, 100}]
FindRoot::cvnwt: Newton's method failed to converge to the prescribed accuracy
```

after 15 iterations.

(在 15 次迭代后,牛顿方法也不能收敛到指定精度.)

{x→ − 0.000774542}

缺省情况中,FindRoot 在退出之前要进行 15 次迭代.在退出之前进行的迭代次数由选项 **MatIterations** 确定.

- **MaxIterations →n** 要求 Mathematica 在退出求解算法之前最多进行 n 次迭代.

例 18 在前面那个例子中,进行 25 次迭代,从而得到更好的结论.

```
FindRoot[x + Sin[x] == 0, {x, 100}, MaxIterations →25]
{x→7.744317860602923 × 10^-11}
```

FindRoot 会尽力发现实根.然而,如果指定了复的初始值,或者方程中包含复数,那么就会给出复根.注意下面这个例子中的方程没有实根.

例 19 **FindRoot[x² + x + 1 == 0, {x, 1}]**

FindRoot::cvnwt: Newton's method failed to converge to the prescribed accuracy after 15 iterations.

(在 15 次迭代后,牛顿方法也不能收敛到指定精度.)

FindRoot[x² + x + 1 == 0, {x, I}]

{x→ − 0.5 + 0.866025 i}

如果牛顿方法不能解决给定的问题,那么割线方法就是一个很好的替代选择.[①]

例 20 方程 $\sqrt{|x|} + x - 1 = 0$ 在 0 与 1 之间有一个根,然而由于 Mathematica 不能计算左边函数的导数,因此牛顿方法失效.这时只要指定两个初始值就会让 Mathematica 使用割线方法计算方程的根.

FindRoot[√Abs[x] + x − 1, {x, 1}]

FindRoot::frjc : Could not symbolically find the Jacobian of √Abs[x] + x − 1. Try giving two starting values for each variable.

(不能符号计算 √Abs[x] + x − 1 的雅可比矩阵,因此不妨对每个变量给出两个初始值试一下.)

FindRoot[√Abs[x] + x − 1, {x, {0, 1}}]

{x→0.381966}

另外,有时虽然函数的导数不能自动确定,但是可以利用 **Jacobian** 选项使得牛顿方法成功地计算出根.对于一元函数,雅可比矩阵就是普通导数.

- **Jacobian →导数**"指定"在算法中所用导数的定义.默认值为 **Jacobian →Automatic**,它要求在有可能的情况下,符号计算出来导数.

例 21 $\sqrt{|x|}$ 的导数为 $\frac{\operatorname{sign}(x)}{2\sqrt{|x|}}$,其中 $\operatorname{sign}(x) = \begin{cases} 1, 如果\ x > 0, \\ -1, 如果\ x < 0 \end{cases}$.(验证这一导数公式!)

FindRoot[√Abs[x] + x − 1, {x, 1}, Jacobian → sign[x]/(2 √Abs[x]) + 1]

{x→0.381966}

在 FindRoot 命令和其它的数值算法中,有两个选项控制运算的精度.

---

① 牛顿方法使用切线方向的 x 截距提高初始估计值的准确度,因此,如果不能计算函数的导数,那么牛顿方法就会失效.虽然割线方法速度有些慢,但它使用的是函数在两个不同点的值,由此计算割线的 x 截距.

- **WorkingPrecision** 选项指定在内部计算时保持多少位的精确度.(在大多数计算机上的)默认值是 **WorkingPrecision**→16.
- **AccuracyGoal** 选项指定所得结果应具有多少位准确有效数字.如果|lhs-rhs|<AccuracyGoal,那么迭代就会结束.默认值为 **AccuracyGoal →Automatic**,它是比 WorkingPrecision 小 10 的数.

例 22　我们希望给出方程 $\cos\left(\frac{100}{x}\right)=\frac{x}{x+1}$ 最靠近 5000 的近似解,准确到第 10 位小数.

```
FindRoot[Cos[100/x] == x/(x+1), {x, 5000}]
{x→5000.}
```

这说明 Mathematica 的默认值不能给出所要求的准确度.增加 WorkingPrecision 的值,就可以达到所要求的精度.

```
FindRoot[Cos[100/x] == x/(x+1), {x, 5000}, WorkingPrecision→24]
{x→5000.83319115955609589817}
```

在默认情况中,AccuracyGoal 是比 WorkingPrecision 小 10 的数,因此只有前 14 位数字是可信的,即 $x\approx5000.8331911596$.

对牛顿方法稍进行改动,就可以进行中间计算,并显示中间计算过程的结果.所用方法就是在要求解的方程左边加上额外的指令,指令间用分号隔开.在下面两个例子中就演示了用这种方法求解方程 $e^{-x}=x$ 的情形.

例 23　为了查看在求解方程 $e^{-x}=x$ 的过程中近似解序列的收敛速度,我们可以要求 **FindRoot** 打印出中间计算的结果.

```
n = -1;
FindRoot[If[n≥0, Print[n," ", x]]; n++; Exp[-x] == x, {x, 2}]
0    2.
1    0.357609
2    0.558708
3    0.56713
4    0.567143
{x→0.567143}
```

例 24　为了比较牛顿方法与割线方法,我们设置 WorkingPrecision→100,利用 FindRoot 打印出达到收敛要求时的迭代次数,并使用 Timing 确定程序花费的时间(具体时间当然与所用计算机速度有关).

```
n = 0;
FindRoot[n++;Exp[-x] == x, {x, 1}, WorkingPrecision→100]//Timing
n
{0.11 Second,
  x→0.5671432904097838729999686622103555497538157871865125081
     351310792230457930866845666932194469617522946}}
9
```

牛顿方法:9 次迭代,0.11 秒.

```
n = 0;
FindRoot[n++;Exp[-x] == x, {x,1,2},WorkingPrecision→100]//Timing
```

```
{0.33 Second,
x→0.5671432904097838729999686622103555497538157871865125081351310792230457930866845666932194469617522946}}
n
28
```

割线方法:28 次迭代,0.33 秒.

如果所求解的方程有一个根的重数为 2 或更多,那么牛顿方法就会收敛得很慢,或者根本不收敛.对于这种情况,通过武断地选择一个 **DampingFactor** 可能会使收敛得到改进.

- **DampingFactor** →因子选项控制在牛顿算法中的收敛行为.在牛顿方法中的每一步的大小都要乘以因子的值.默认值为 **DampingFactor** →1.

例 25 **FindRoot[(Exp[x] - 1)^2, {x,2}]**　　←0 为二重零点.

FindRoot::cvnwt: Newton's method failed to converge to the prescribed accuracy after 15 iterations.

(在 15 次迭代后,牛顿方法也不能收敛到指定精度.)

{x→0.000911355}

FindRoot[(Exp[x] - 1)^2, {x, 2}, DampingFactor→2]

{x→7.73452×10^{-6}}

虽然这里没有显示,实际上进行了八次迭代后就收敛到指定要求.

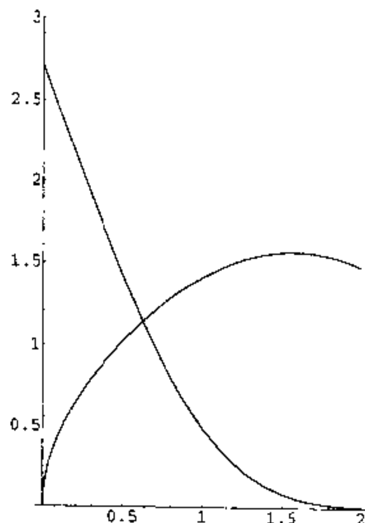
FindRoot 也可以用来确定联立方程组的解.

■ **FindRoot[方程组, {变量 1, a1}, {变量 2, a2}, …]** 当变量 1, 变量 2, …的初始值分别为 a1, a2, … 时,尝试求解方程组.这里的方程组形式为{方程 1,方程 2,…}.另外,方程与方程之间也可以用 && (逻辑与)分开.

求解多元函数的牛顿方法与一元时的情形不同,它对初始值的选择相当敏感.因此,这时函数的图形就是相当有用的帮助.

例 26 求解方程组 $\begin{cases} e^x + \ln y = 2, \\ \sin x + \sin y = 1. \end{cases}$

首先画出两个方程的图形.



当执行这个命令时,会看到一些警告信息.这时可以不必理会它们.

≪Graphics`ImplicitPlot`

ImplicitPlot[{Exp[x] + Log[y] == 2, Sin[x] + Cos[y] == 1}, {x,0,2}, PlotRange

```
→{0, 3}]
```

由此可见,只有惟一解.令 $x=1, y=1$ 为初始估计值.

```
FindRoot[{Exp[x] + Log[y] == 2, Sin[x] + Cos[y] == 1}, {x, 1}, {y, 1}]
{x→0.624295, y→1.14233}
```

如果方程中有一个函数,计算它的值要花费很多时间,尤其是当需要高精度解时更为如此,那么下面这个程序就相当有用.

■ **InterpolateRoot[*lhs* == *rhs*, {x, a, b}]** 利用初始值 a 与 b 求解方程 *lhs* == *rhs*.

同 FindRoot 一样,这里也可以用一个函数代替方程,这时计算的是函数的零点.

与 FindRoot 用线性函数(直线)近似方程的根不同,InterpolateRoot 利用三次或更低次多项式近似方程的根.这样就导致只要计算更少的函数值,就可以得到更高精度的解.InterpolateRoot 包含在软件包 **NumericalMath`InterpolateRoot`** 中,因此在使用命令前,要上载这个软件包.

例 27 这个例子利用工作精度为 **1000** 位有效数字计算贝塞尔函数[①] $J_0(x)$ 的介于 2 与 3 之间的零点.为了进行比较,这里使用了 Mathematica 函数 Timing.另外为了节省空间,这里没有显示计算所得的近似值.(去掉指令后面的分号,再执行指令,就会看到计算的结果.)

```
FindRoot[BesselJ[0, x], {x, 2}, WorkingPrecision→1000]; //Timing
{7.69 Second, Null}
<<NumericalMath`InterpolateRoot`
InterpolateRoot[BesselJ[0, x], {x, 2, 3}, WorkingPrecision→1000]; //Timing
{4.95 Second, Null}
```

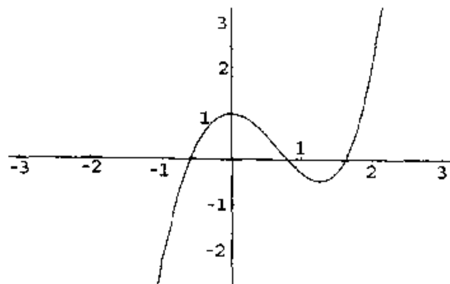
习 题 解 答

6.11 求解方程 $5\cos x = 4 - x^3$,确保求出了所有的解.

解 ☞

由于 $5\cos x = 4 - x^3$ 当且仅当 $5\cos x - 4 + x^3 = 0$,所以引进函数 $f(x) = 5\cos x - 4 + x^3$,并查看它的 x 截距.(虽然我们可以查看两条曲线的交点,但这里所用的方法更容易确定交点在坐标轴上的位置.)

```
f[x_] = 5 Cos[x] - 4 + x^3;
Plot[f[x], {x, -3, 3}];
```



由此可见,方程有三个解,分别邻近 $-0.5, 0.8$ 与 1.6.

```
FindRoot[f[x], {x, -.5}]
{x→-0.576574}
FindRoot[f[x], {x, 0.8}]
{x→0.797323}
FindRoot[f[x], {x, 1.6}]
{x→1.61805}
```

6.12 求出方程[②] $\sin x = 2$ 的一个解.

① $J_0(x)$ 为微分方程 $x^2y'' + xy' + x^2y = 0$ 的一个解.

② 如果你不熟悉复变量函数,那么不妨跳过这个习题.

解 ☞

由于对所有的实数 x, $-1\leqslant\sin x\leqslant1$,所以这个问题没有实数解.这时我们可以通过给定一个复数初始值,强迫 FindRoot 计算复数解.

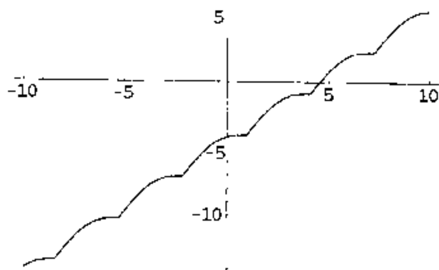
```
FindRoot[Sin[x]==2, {x, I}]
{x→1.5708 + 1.31696 i}
```

6.13 给出方程 $x+|\sin(x-1)|=5$ 的有 20 位有效数字的近似解.

解 ☞

首先画出 $f(x)=x+|\sin(x-1)|-5$ 的图形.

```
f[x_] = x + Abs[Sin[x-1]]-5
Plot[f[x], {x, -10, 10}]
```



由此可见,方程只在 4 与 5 之间有一个解.

```
FindRoot[f[x], {x, 5}]
FindRoot::frjc : Could not symbolically find the Jacobian of {f[x]}. Try giving two starting values for each variable.
```

(不能符号计算{f[x]}的雅可比矩阵,因此不妨对每个变量给出两个初始值试一下.)

由于 Mathematica 无法确定 $f(x)$的导数,所以牛顿方法失效.这时可以尝试 **WorkingPrecision→30** (即比要求精度多 10 位,因为默认情况中,AccuracyGoal 比 WorkingPrecison 少 10)时的割线方法.

```
FindRoot[f[x], {x, 4, 5}, WorkingPrecision→30]
{x→4.57764001198757729525937359564}
```

保留到 20 位有效数字,可知方程的解为 4.5776400119875772953.

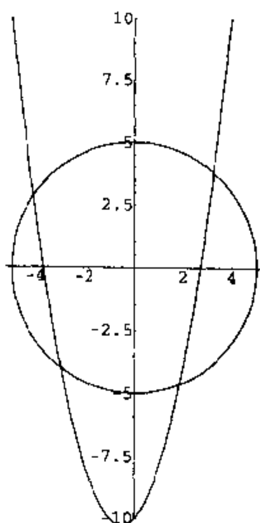
6.14 求出抛物线 $y=x^2+x-10$ 与圆 $x^2+y^2=25$ 的交点.

解 ☞

首先画出它们的图形.

```
g1 = Graphics[Circle[{0, 0}, 5]];
g2 = Plot[x^2 + x - 10, {x, -5,5},
DisplayFunction→Identity];
Show [g1, g2, AspectRatio→Automatic,
     PlotRange→{-10, 10}, Axes→True,
     DisplayFunction→$DisplayFunction];
```

抛物线 $y=x^2+x-10$ 与圆 $x^2+y^2=25$ 有四个交点.下面求解这些交点的坐标.由于精确解结构非常复杂,所以给出数值近似解.



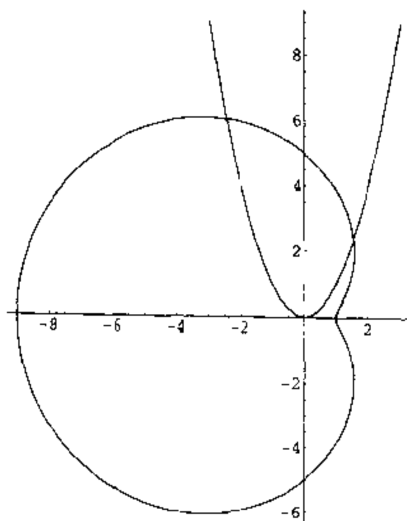

```
NSolve[y == x^2 + x - 10 && x^2 + y^2 == 25]
{{y→2.83654, x→-4.11753}, {y→3.80098, x→3.24846}, {y→-4., x→-3. },
{y→-4.63752, x→1.86907}}
```

6.15 求出蚶线 $r=5-4\cos\theta$ 与抛物线 $y=x^2$ 的交点.

解 ☞

首先在同一坐标系中画出两条曲线的图形.

```
<<Graphics`Graphics`
limacon = PolarPlot[5 - 4Cos[t], {t, 0, 2π},DisplayFunction→Identity];
parabola = Plot[x^2, {x, -3,3}, DisplayFunction→Identity];
Show[limacon, parabola, DisplayFunction→$DisplayFunction];
```



下面把蚶线的方程转化为直角坐标:

$$r=5-4\cos\theta$$

$$r^2=5r-4r\cos\theta \qquad \boxed{\begin{array}{l} r=\sqrt{x^2+y^2} \\ x=r\cos\theta \end{array}}$$

$$x^2+y^2=5\sqrt{x^2+y^2}-4x$$

第一个交点靠近 (2, 2)：

```
FindRoot[{y == x^2, x^2 + y^2 == 5 Sqrt[x^2 + y^2] - 4x}, {x,2}, {y,2}]
{x→1.53711, y→2.3627}
```

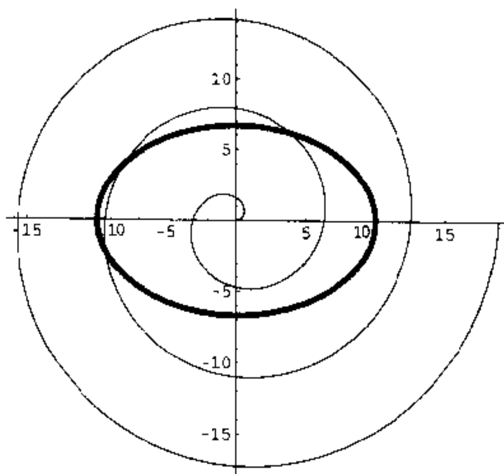
第二个交点靠近 (−3, 6)：

```
FindRoot[{y == x^2, x^2 + y^2 == 5 Sqrt[x^2 + y^2] - 4x}, {x, -3}, {y, 6}]
{x→ - 2.4552, y→6.02802}
```

6.16 阿基米德螺线 $r=\theta$ 与椭圆 $4x^2+9y^2=400$ 交于哪些点？

解 ☞

```
<<Graphics`Graphics`
<<Graphics`ImplicitPlot`
spiral = PolarPlot[θ, {θ, 0, 6π}, DisplayFunction →Identity];
ellipse = ImplicitPlot[ 4x^2 + 9y^2 == 400, {x, -10, 10},
                       PlotStyle →Thickness[.01],
                       DisplayFunction →Identity];
Show[spiral, ellipse, DisplayFunction → $DisplayFunction];
```



由上图可见，交点靠近 (4, 6)，(−8, 4) 与 (−9, −2) 点．为了把极坐标转化为直角坐标，应用变换 $r=\sqrt{x^2+y^2}$ 和 $\theta=\tan^{-1}(y/x)$．这样就得到方程 $\sqrt{x^2+y^2}=\tan^{-1}(y/x)$．然而，如果应用 $\tan(\sqrt{x^2+y^2})=y/x$ 的话，牛顿方法就更稳定．

```
FindRoot[{Tan[Sqrt[x^2 + y^2]] == y/x, 4x^2 + 9y^2 == 400}, {x, 4}, {y, 6}]
{x→3.93476, y→6.1289}
FindRoot[{Tan[Sqrt[x^2 + y^2]] == y/x, 4x^2 + 9y^2 == 400}, {x, -8}, {y, 4}]
{x→ - 8.04703, y→3.95785}
FindRoot[{Tan[Sqrt[x^2 + y^2]] == y/x, 4x^2 + 9y^2 == 400}, {x, -9}, {y, -2}]
{x→ - 9.38786, y→ - 2.29668}
```

6.17 求出下述方程组靠近点 (1, 2, 3) 的一个解：

$$\begin{cases} x + y + z = 6, \\ \sin x + \cos y + \tan z = 1, \\ e^x + \sqrt{y} + \dfrac{1}{z} = 5. \end{cases}$$

解 ☞

```
FindRoot[{x + y + z == 6, Sin[x] + Cos[y] + Tan[z] == 1,
    Exp[x] + √y + 1/z == 5}, {x, 1}, {y, 2}, {z, 3}]
{x→1.23382, y→1.5696, z→3.19658}
```

第七章　代数与三角

7.1　多项式

由于在代数中非常普遍地用到多项式,因此 Mathematica 专门提供了处理多项式的命令.

- **PolynomialQ[表达式, 变量]** 如果表达式为关于变量的多项式,返回 True;否则返回 False.如果在表达式中只出现了一个变量,可以不给出变量参数.
- **Variables[多项式]** 给出在多项式中所有独立变量组成的列表.
- **Coefficient[多项式, 项]** 给出在多项式中项的系数.**Coefficient[多项式, 项, n]** 给出在多项式中项的 n 次幂的系数.
- **CoefficientList[多项式, 变量]** 给出在多项式中变量各个幂次的系数组成的列表,从第 0 次幂开始.

例 1

```
PolynomialQ[x^2 + 3x + 2]
True
PolynomialQ[x^2 + 3x + 2/x]
False
PolynomialQ[x^2 + 3x + 2/y, x]        ←对 x 而言,2/y 为常数.
True
```

例 2

```
poly1 = (x + 1)^10;
poly2 = x^3 - 5x^2 y + 3 x y^2 - 7y^3;
Variables[poly2]
{x, y}
Coefficient[poly1, x, 5]
252
Coefficient[poly2, x]
3y^2
Coefficient[poly2, y, 2]
3x
Coefficient[poly2, x y^3]
3
CoefficientList[poly1, x]
{1, 10, 45, 120, 210, 252, 210, 120, 45, 10, 1}
CoefficientList[poly2, x]
{-7y^3, 3y^2, -5y, 1}
CoefficientList[poly2, y]
{x^3, -5x^2, 3x, -7}
```

有时候,把多项式方程的解写作逻辑表达式是非常方便的.例如,如果 $x^2 - 4 = 0$,那么 $x = -2$ 或者 $x = 2$.可以用两条特殊的命令 **Roots** 与 **NRoots** 把多项式方程的根按这种方法表示出来.此时解用符号||(逻辑或)表示成互不相交的形式.

- Root[*lhs* == *rhs*, 变量] 给出多项式方程的解.
- NRoot[*lhs* == *rhs*, 变量] 给出多项式方程的数值近似解.

例 3 求出方程 $x^4+x^3-8x^2-5x-15=0$ 的所有大于 2 的解.

```
solutions = Roots[x^4 + x^3 - 8x^2 - 5x + 15 == 0, x]
```

$x == \frac{1}{2}(-1-\sqrt{13}) || x == \frac{1}{2}(-1+\sqrt{13}) || x == \sqrt{5} || x == -\sqrt{5}$

```
solutions && x > 2 //Simplify
```

$x == \sqrt{5}$

&& 为 Mathematica 的逻辑与. 见第 7.4 节关于 Simplify 的讨论.

```
numericalsolutions = NRoots[x^4 + x^3 - 8x^2 - 5x + 15 == 0, x]
x == -2.30278 || x == -2.23607 || x == 1.30278 || x == 2.23607
numericalsolutions && x > 2 // Simplify
x == 2.23607
```

多项式的除法保证任意给定两个多项式 p 与 s,其中 $\deg(p) \geqslant (\deg(s)$,都存在惟一确定的多项式 q 与 r,使得

$$p(x) = q(x)\, s(x) + r(x), \qquad 其中\ \deg(r) < \deg(s).$$

在 Mathematica 中生成商与余式的命令为

- **PolynomialQuotient[p, s, x]** 给出 p 除以 s 所得的商式,表示成 x 的函数. 余式被忽略.
- **PolynomialRemainder[p, s, x]** 返回 p 除以 s 所得的余式. 余式的次数小于 s 的次数.

例 4

```
p = x^5 - 7x^4 + 3x^2 - 5x + 9;
s = x^2 + 1;
q = PolynomialQuotient[p, s, x]
10 - x - 7x^2 + x^3
r = PolynomialRemainder[p, s, x]
-1 - 4x
```

- **Expand[多项式]** 展开乘积与幂,把多项式写成单个项的和的形式.
- **Factor[多项式]** 尝试在整数环上分解多项式. 如果不能分解的话,多项式不发生变化.
- **FactorTerms[多项式]** 从多项式中分解出来每项的公共常数. **Factor[多项式, 变量]** 会分解出不包含变量的公共单项式.
- **Collect[多项式, 变量]** 处理有两个或更多变量的多项式,把它表示成变量的多项式.

例 5

```
poly = 6x^2y^3z^4 + 8x^3y^2z^5 + 10x^2y^4z^3;
Factor[poly]
2x^2y^2z^3(5y^2 + 3yz + 4xz^2)          ←poly 被完全分解.
FactorTerms[poly, x]
2y^2z^3(5x^2y^2 + 3x^2yz + 4x^3z^2)     ←只有不包含 x 的项被分解出来.
FactorTerms[poly, y]
2x^2z^3(5y^4 + 3y^3z + 4x y^2z^2)       ←只有不包含 y 的项被分解出来.
```

```
FactorTerms[poly, z]
2x^2y^2(5y^2z^3 + 3yz^4 + 4xz^5)
```
←只有不包含 z 的项被分解出来.

例 6
```
poly = 1 + 2x + 3y + 4 x y + 5 x^2 y + 6x y^2 + 7x^2 y^2;
Collect[poly, x]
1 + 3y + x(2 + 4y + 6y^2) + x^2(5y + 7y^2)
```
← x 的幂次被整理出来.
```
Collect[poly, y]
1 + 2x + (3 + 4x + 5x^2)y + (6x + 7x^2)y^2
```
← y 的幂次被整理出来.

默认情况下,Factor 只会在整数环上分解因式.有一个选项可以改变这一点.

- **Extension → {扩展 1, 扩展 2,…}** 用来指定可以使用的一组代数数.如果只有一个扩展,那么不必加上大括号{}.**Extension →Automatic 表示扩展到包含所有出现在多项式中的代数数全体的域上.**
- **GaussianIntegers →True** 允许在包含i的整数集上进行分解因式.另外,在扩展集中也可以包含i或 I.

例 7
```
Factor[x^8 - 41x^4 + 400]
(-2 + x)(2 + x)(-5 + x^2)(4 + x^2)(5 + x^2)
Factor[x^8 - 41x^4 + 400, GaussianIntegers →True]
(-2 + x)(-2 i + x)(2 i + x)(2 + x)(-5 + x^2)(5 + x^2)
Factor[x^8 - 41x^4 + 400, Extension →√5]
-(√5 - x)(-2 + x)(2 + x)(√5 + x)(4 + x^2)(5 + x^2)
Factor[x^8 - 41x^4 + 400, Extension →{I, √5 }]
-(√5 - x)(√5 - i x)(√5 + i x)(-2 + x)(-2 i + x)(2 i + x)(2 + x)(√5 + x)
```

多项式 p_1, p_2,… 的最大公因式(GCD)就是能同时整除 p_1, p_2,…的最高次数多项式. p_1, p_2,… 的最小公倍式(LCM)就是能同时被 p_1, p_2,…整除的最低次数多项式.

- **PolynomialsGCD[p1, p2, …]** 计算出多项式 p1, p2, …的最大公因式.
- **PolynomialsLCM[p1, p2, …]** 计算出多项式 p1, p2, …的最小公倍式.

例 8
```
p = (x - 1)(x - 2)^2(x - 3)^3;
q = (x - 1)^2(x - 2)(x - 3)^4;
PolynomialGCD[p, q]
(-3 + x)^3(-2 + x)(-1 + x)
PolynomialLCM[p, q]
(-3 + x)^4(-2 + x)^2(-1 + x)^2
```

缺省情况下,PolynomialGCD 与 PolynomialLCM 假定多项式的系数为有理数.同 Factor 一样,Extension 选项可以用来指定一组代数数(可以包含 1),供计算时使用.

例 9
```
p = x^2 - 5;
q = x + √5;
PolynomialGCD[p, q]
1
```

```
PolynomialLCM[p, q]
(√5 + x)( - 5 + x^2)            ←不使用 Extension →Automatic,认为√5为单独变量.
PolynomialGCD[p, q, Extension →Automatic]
√5 + x
PolynomialLCM[p, q, Extension →Automatic]
( - 5 + x^2)
```

虽然 Mathematica 会自动展开乘积与商的整数幂次,但是如果指数非整数,那么表达式就不会变化.为了强迫指数被“分配”开,可以使用 **PowerExpand** 命令.

■ **PowerExpand[表达式]** 展开积与商,它们的开方以及对数的嵌套幂次.

例 10

```
(a b)^5
a^5 b^5                        ←由于指数为整数,所以 Mathematica 把它分配开.
(a b)^x
(a b)^x                        ←由于指数未定义,所以 Mathematica 不做任何操作.
PowerExpand[(a b)^x]
a^x b^x                        ←利用 PowerExpand 强制进行展开.
```

当使用了多值函数时,必须仔细使用 PowerExpand 命令.

例 11

```
√(a b)/. {a → - 1, b → - 1}
1
PowerExpand[ √(a b)]
√a√b
PowerExpand[ √(a b) /. {a → - 1, b → - 1}
 - 1                           ←先用 PowerExpand 进行展开,然后代入 a, b 的值 - 1.
```

下面是演示 PowerExpand 用法的另外几个例子.

例 12

```
(a^x)^y// PowerExpand
a^(x y)
(a/b)^x// PowerExpand
a^x b^-x
Log[x y] // PowerExpand
Log[x] + Log[y]
Log[x/y] // PowerExpand
Log[x] - Log[y]
Log[x^y] // PowerExpand
y Log[x]
```

习 题 解 答

7.1 判断 $1 + x\sin y + x^2\cos y + x^5 e^y$ 是否为 x 的多项式.它是 y 的多项式吗?

解 ☞

```
PolynomialQ[1 + x Sin[y] +
    x^2 Cos[y] + x^5 Exp[y], x]        ←在这个表达式中认为 y 为常数.
True
PolynomialQ[1 + x Sin[y] + x^2 Cos[y] + x^5 Exp[y], y]
False
```

7.2 展开多项式 $(2x+3)^5$ 后系数都是多少?

解 ☞

```
poly = (2x + 3)^5;
CoefficientList[poly, x]
{243, 810, 1080, 720, 240, 32}
```

7.3 在展开 $(x+y+z)^6$ 后,$x\,y^2\,z^3$的系数是多少?

解 ☞

```
poly = (x + y + z)^6;
CoefficientList[poly, x y^2 z^3]
60
```

7.4 完全展开 $(x+a+1)^4$.

解 ☞

```
Expand[(x + a + 1)^4]
1 + 4a + 6a^2 + 4a^3 + a^4 + 4x + 12ax + 12a^2x + 4a^3x + 6x^2 + 12ax^2 + 6a^2x^2 + 4x^3 + 4ax^3 + x^4
```

7.5 把 $(x+a+1)^4$表示成 x 的多项式.

解 ☞

```
Collect[(x + a + 1)^4, x]
1 + 4a + 6a^2 + 4a^3 + a^4 + (4 + 12a + 12a^2 + 4a^3)x + (6 + 12a + 6a^2)x^2 + (4 + 4a)x^3 + x^4
```

7.6 分解多项式

$$poly = 6x^3 + x^2y - 11xy^2 - 6y^3 - 5x^2z + 11xyz + 11y^2z - 2xz^2 - 6yz^2 + z^3$$

并相应于 z 求解方程 $poly=0$.

解 ☞

```
poly = 6x^3 + x^2 y - 11x y^2 - 6 y^3 - 5 x^2 z + 11x y z + 11y^2 z - 2x z^2 - 6yz^2 + z^3;
Factor[poly]
(x + y - z)(3x + 2y - z)(2x - 3y + z)
Roots[poly == 0, z]
z == x + y || z == 3x + 2y || z == -2x + 3y
```

7.7 计算 $x^5+2x^4-3x^3+7x^2-10x+5$ 除以 x^2-4 的商与余式,并验证答案的正确性.

解 ☞

```
p = x^5 + 2x^4 - 3x^3 + 7x^2 - 10x + 5;
```

```
s = x^2 - 4
q = PolynomialQuotient[p, s, x]
15 + x + 2x^2 + x^3
r = PolynomialRemainder[p, s, x]
65 - 6x
checkpoly = q * s + r //Expand
5 - 10x + 7x^2 - 3x^3 + 2x^4 + x^5
checkpoly == p
True
```

7.8 把 $(x+y+z)^3$ 表示成 z 的多项式.

解 ☞

```
Collect[(x + y + z)^3, z]
x^3 + 3x^2y + 3xy^2 + y^3 + (3x^2 + 6xy + 3y^2)z + (3x + 3y)z^2 + z^3
```

7.9 令 $p=2x^4-15x^3+39x^2-40x+12$, $q=4x^4-24x^3+45x^2-29x+6$, 计算它们的最大公因式与最小公倍式,并证明所得结果的乘积等于 pq.

解 ☞

```
p = 2x^4 - 15x^3 + 39x^2 - 40x + 12;
q = 4x^4 - 24x^3 + 45x^2 - 29x + 6;
a = PolynomialGCD[p, q]
-6 + 17x - 11x^2 + 2x^3
b = PolynomialLCM[p, q]
(-2 + x)(6 - 29x + 45x^2 - 24x^3 + 4x^4)
Expand[a * b] == Expand[p * q]
True
```

7.10 在整数环以及包含$\sqrt{5}$与 I 的数域上分解 x^4-25.

解 ☞

```
Factor[x^4 - 25]
(-5 + x^2)(5 + x^2)
Factor[x^4 - 25, Extension -> {√5, I}]
-(√5 - x)(√5 - i x)(√5 + i x)(√5 + x)
```

7.11 展开 $\ln\left(\sqrt{\dfrac{x^a y^b}{z^c}}\right)$.

解 ☞

```
Log[Sqrt[x^a y^b / z^c]] //PowerExpand
1/2 (a Log[x] + b Log[y] - c Log[z])
```

7.2 有理函数与代数函数

有几条命令适合于处理有理函数(分子与分母均为多项式的分式函数).

- **Numerator[分式]** 返回分式的分子.
- **Denominator[分式]** 返回分式的分母.
- **Cancel[分式]** 消去分式中分子与分母的公因子. 选项 **Extension →Automatic** 使得操作在出现分数中的代数数上进行.
- **Together[表达式]** 利用公分母把表达式中的项合并起来,并且消去分子与分母中的公因子.
- **Apart[分式]** 把分式表示成部分分式的和.

例 13　**Cancel**$\left[\frac{x^2+5x+6}{x^2+3x+2}\right]$

$\frac{3+x}{1+x}$

例 14　**Together**$\left[\frac{1}{x+1}+\frac{2}{x^2-1}\right]$

$\frac{1}{-1+x}$

例 15　**Apart**$\left[\frac{x^2+5x}{x^4+x^3-x-1}\right]$

$\frac{1}{-1+x}+\frac{2}{1+x}+\frac{-1-3x}{1+x+x^2}$

由于在默认情况下,Mathematica 会把负指数转化为等价的正指数,因此 Numerator 与 Denominator 的结果可能与期望的不同.

例 16　**frac** = $\frac{x^{-1}y^{-2}}{z^{-3}}$

Numerator[frac]

z^3

Denominator[frac]

xy^2

- **ExpandNumerator[表达式]** 把表达式中的分子展开,而分母保持不变.
- **ExpandDenominator[表达式]** 把表达式中的分母展开,而分子保持不变.
- **ExpandAll[表达式]** 把表达式的分子与分母都展开,并把结果写成有公分母分式的和.

例 17　**expr** = $\frac{(x+1)\ (x+2)}{(x+3)\ (x+4)}$;

ExpandNumerator[expr]

$\frac{2+3x+x^2}{(3+x)\ (4+x)}$

ExpandDenominator[expr]

$\frac{(1+x)\ (2+x)}{12+7x+x^2}$

ExpandAll[expr]

$\frac{2}{12+7x+x^2}+\frac{3x}{12+7x+x^2}+\frac{x^2}{12+7x+x^2}$

```
ExpandNumerator[ExpandDenominator[expr]]
```

$$\frac{2+3x+x^2}{12+7x+x^2}$$

本节描述的命令并不是只适用有理函数(多项式的商),对于那些包含根式的代数表达式,以及包含函数或者未定义对象的非代数表达式也同样适用.另外,如果在命令中设置了选项 `Trig→True`,那么 Mathematica 就会应用标准的三角函数恒等式简化表达式.在 7.3 节中将对此详加讨论.

例 18 `Expand[(1+√x)^6]`

$$1+6\sqrt{x}+15x+20x^{3/2}+15x^2+6x^{5/2}+x^3$$

例 19 `Apart[1/((√x+1)(√x+2))]`

$$\frac{1}{1+\sqrt{x}}-\frac{1}{2+\sqrt{x}}$$

习 题 解 答

7.12 在微积分中表达式$\frac{f(x)-f(a)}{x-a}$与导数有密切关系.对于 $f(x)=x^9$, $a=-3$ 简化这个表达式.

解 ☞

```
f[x_] = x^9;
a = -3;
Cancel[(f(x) - f(x))/(x - a)]
```

$$6561-2187x+729x^2-243x^3+81x^4-27x^5+9x^6-3x^7+x^8$$

7.13 把$\frac{a}{b}$,$\frac{c}{d}$与$\frac{e}{f}$组合成一个分式.

解 ☞

```
Together[a/b + c/d + e/f]
```

$$\frac{bde+bcf+adf}{bdf}$$

7.14 展开$\frac{(x+2)(x^2+3)(2x-7)}{(x^2+5x+2)(x-5)(x+6)}$的分子和分母.

解 ☞

```
ExpandNumerator[ExpandDenominator[((x+2)(x^2+3)(2x-7))/((x^2+5x+2) (x-5) (x+6))]]
```

$$\frac{-42-9x-8x^2-3x^3+2x^4}{-60-148x-23x^2+6x^3+x^4}$$

```
ExpandAll[((x+2) (x^2+3)(2x-7))/((x^2+5x+2) (x-5) (x+6))]//Together
```

$$\frac{-42-9x-8x^2-3x^3+2x^4}{-60-148x-23x^2+6x^3+x^4}$$

7.15　把$\frac{2x+3}{5x-7}$,$\frac{7x-2}{3x+1}$与$\frac{x^2}{x^2+1}$加起来,并把结果表示成一个分式,分子与分母均为展开形式.

解 ☞

```
p = (2x + 3)/(5x - 7);
q = (7x - 2)/(3x + 1)
r = x^2/(x^2 + 1);
Together[p + q + r]//ExpandDenominator
```

$$\frac{17-48x+51x^2-64x^3+56x^4}{-7-16x+8x^2-16x^3+15x^4}$$

> 如果不使用//ExpandDenominator,分母就会表示成因式分解的形式.

7.16　$\frac{(x-1)^6}{(x^2+1)(x+1)^2(x-4)}$的部分分式展开是什么?

解 ☞

```
Apart[(x - 1)^6/((x^2 + 1)(x + 1)^2(x - 4))]
```

$$-4+\frac{729}{425(-4+x)}+x-\frac{32}{5(1+x)^2}+\frac{288}{25(1+x)}-\frac{4(4+x)}{17(1+x^2)}$$

7.17　把前一习题中的分式展成具有线性复分母的部分分式的和.

解 ☞

```
Apart[(x - 1)^6/((x + I) (x - I) (x + 1)^2(x - 4))]
```

> 为了强迫 Mathematica 用线性复分母表示结果,把 x^2+1 分解为$(x+I)(x-I)$.

$$-4+\frac{729}{425(-4+x)}+x-\frac{\frac{2}{17}-\frac{8i}{17}}{-i+x}-\frac{\frac{2}{17}+\frac{8i}{17}}{i+x}-\frac{32}{5(1+x)^2}+\frac{288}{25(1+x)}$$

7.18　把$(e^x+e^{2x})^4$表示成指数函数的和.

解 ☞

```
Expand[(E^x + E^(2x))^4]
```

$$e^{4x}+4e^{5x}+6e^{6x}+4e^{7x}+e^{8x}$$

7.3　三角函数

虽然在前一节讨论的命令也适用于三角函数,但这样做并没有利用三角函数恒等式进行简化.为了把这些恒等式结合到运算过程中,必须设置选项 **Trig →True**.(对于除 Simplify 外的所有命令而言,默认值为 Trig→False.)下面的例子演示了其中的差别.

例 20

```
Cancel[Sin[x]/(1 - Cos[x]^2)]
```

$$-\frac{\mathrm{Sin}[x]}{-1+\mathrm{Cos}[x]^2}$$

```
Cancel[Sin[x]/(1 - Cos[x]^2), Trig →True]
```

Csc[x]

例 21
```
Together[Cos[x]^2/(1 - Sin[x]^2) + Sin[x]^2/(1 - Cos[x]^2)]
```
$$\frac{\mathrm{Cos}[x]^2-\mathrm{Cos}[x]^4+\mathrm{Sin}[x]^2-\mathrm{Sin}[x]^4}{(-1+\mathrm{Cos}[x]^2)(-1+\mathrm{Sin}[x]^2)}$$
```
Together[Cos[x]^2/(1 - Sin[x]^2) + Sin[x]^2/(1 - Cos[x]^2), Trig -> True]
2
```
Trig -> True 除了适用于圆函数以外,也同样适用于双曲函数.

例 22
```
Expand[(Cosh[x]^2 + Sinh[x]^2)(Cosh[x]^2 - Sinh[x]^2)]
Cosh[x]^4 - Sinh[x]^4
Expand[(Cosh[x]^2 + Sinh[x]^2)(Cosh[x]^2 - Sinh[x]^2), Trig -> True]
Cosh[x]^2 + Sinh[x]^2
```

为了对三角函数进行其它的操作,Mathematica 专门提供了下述命令,这些函数同时适用于圆函数与双曲函数.

- **TrigExpand[表达式]** 利用某种三角恒等式展开表达式.
- **TrigReduce[表达式]** 把表达式中的三角函数的乘积与幂写成具有复合参数的三角表达式,从而把表达式简化为三角函数的线性形式(没有乘积与幂).
- **TrigFactor[表达式]** 把表达式转化为等价的但没有角度的和与倍数的表达式,然后对所得三角表达式分解因式.

下面的例子说明了 Expand 与 TrigExpand 之间的差别.

例 23
```
Expand[(Sin[x] + Cos[x])^2]
Cos[x]^2 + 2 Cos[x]Sin[x] + Sin[x]^2
TrigExpand[(Sin[x] + Cos[x])^2]
1 + 2 Cos[x]Sin[x]
```

例 24
```
TrigExpand[Sin[x + y]]
Cos[y]Sin[x] + Cos[x] Sin[y]
TrigExpand[Sin[2x]]
2 Cos[x]Sin[x]
TrigExpand[Sin[2x + y]]
2 Cos[x] Cos[y] Sin[x] + Cos[x]^2 Sin[y] - Sin[x]^2 Sin[y]
```

TrigExpand 也适用于双曲函数.

例 25
```
TrigExpand[Cosh[x + y]]
Cosh[x] Cosh[y] + Sinh[x] Sinh[y]
```

例 26
```
TrigReduce[Sin[2x]^2 + Sin[x] Cos[3x]^3]
```

> **TrigReduce** 把原来的表达式表示成线性三角表达式.

$$\frac{1}{8}(4 - 4\,\mathrm{Cos}[4x] - 3\,\mathrm{Sin}[2x] + 3\,\mathrm{Sin}[4x] - \mathrm{Sin}[8x] + \mathrm{Sin}[10x])$$

```
TrigReduce[Sinh[2x]^2 + Sinh[x] Cosh[3x]^3]
```

$\frac{1}{8}(-4+4\,\text{Cosh}[4x]-3\,\text{Sinh}[2x]+3\,\text{Sinh}[4x]-\text{Sinh}[8x]+\text{Sinh}[10x])$

下面的例子说明了 TrigFactor 与 TrigReduce 的差别.注意 TrigFactor 把表达式写成积的形式,而 TrigReduce 把表达式写成和的形式.

例 27

```
expr = TrigExpand[4 Sin[x]^2 Cos[2x]^3]
3 Cos[x]^2 Sin[x]^2 + Cos[x]^6 Sin[x]^2 - 3 Sin[x]^4 -
15 Cos[x]^4 Sin[x]^4 + 15 Cos[x]^2 Sin[x]^6 - Sin[x]^8
TrigFactor[expr]
4(Cos[x] - Sin[x])^3 Sin[x]^2(Cos[x] + Sin[x])^3
TrigReduce[expr]
```

$\frac{1}{4}(-3+6\,\text{Cos}[2x]-4\,\text{Cos}[4x]+2\,\text{Cos}[6x]-\text{Cos}[8x])$

Sovle 命令可以用来求解三角函数方程.然而,由于返回的只是反三角函数的主值,因此得到的并不是全部解.

例 28 考虑方程 $1-2\cos x-\sin x+\sin 2x=0$.

```
equation = 1 - 2 Cos[x] - Sin[x] + Sin[2x] == 0;
Solve[equation, x]
Solve::ifun:
Inverse functions are being used by Solve, so some solutions may not be found.
```

(Solve 使用了反函数,因此有可能有些解没有求出.)

$\left\{\left\{x\to-\frac{\pi}{3}\right\},\left\{x\to\frac{\pi}{3}\right\},\left\{x\to\frac{\pi}{2}\right\}\right\}$

由于三角函数与双曲函数都可以用指数函数表示(对于圆周三角函数,要用到复指数函数),因此 Mathematica 提供了两个转化函数:

- **TrigToExp[表达式]** 把三角函数与双曲函数转化为指数函数形式.
- **ExpToTrig[表达式]** 把指数函数转化为三角函数与(或)双曲函数.

例 29

```
TrigToExp[Cos[x]]
```

$\frac{e^{-ix}}{2}+\frac{e^{ix}}{2}$

```
TrigToExp[Sinh[x]]
```

$-\frac{e^{-x}}{2}+\frac{e^{x}}{2}$

```
ExpToTrig[Exp[x]]
Cosh[x] + Sinh[x]
```

习 题 解 答

7.19 化简三角函数表达式$\frac{1}{\cos^2 x-\sin^2 x}$.

解 ☞

```
TrigReduce[1/(Cos[x]^2 - Sin[x]^2)]
Sec[2x]
```

7.20 简化并分解因式：$\sin^2 x \cos^2 x + \cos^4 x$.

解 ☞

```
TrigFactor[Sin[x]^2 Cos[x]^2 + Cos[x]^4]
Cos[x]^2
```

7.21 求解三角函数方程：$1 - 2\cos x - 2\sin x + 4\sin 2x = 0$.

解 ☞

```
equation = 1 - 2 Cos[x] - 2 Sin[x] + 4 Sin[2x] == 0;
Solve[equation, x]
Solve::ifun:
Inverse functions are being used by Solve, so some solutions may not be found.
```

(Solve 使用了反函数，因此有可能有些解没有求出.)

$$\left\{\left\{x\to \text{ArcCos}\left[\frac{1}{8}+\frac{\sqrt{13}}{8}-\frac{1}{2}\sqrt{\frac{9}{8}-\frac{\sqrt{13}}{8}}\right]\right\},\right.$$

$$\left\{x\to \text{ArcCos}\left[\frac{1}{8}+\frac{\sqrt{13}}{8}+\frac{1}{2}\sqrt{\frac{9}{8}-\frac{\sqrt{13}}{8}}\right]\right\},$$

$$\left\{x\to \text{ArcCos}\left[\frac{1}{8}-\frac{\sqrt{13}}{8}-\frac{1}{2}\sqrt{\frac{9}{8}+\frac{\sqrt{13}}{8}}\right]\right\},$$

$$\left.\left\{x\to -\text{ArcCos}\left[\frac{1}{8}-\frac{\sqrt{13}}{8}+\frac{1}{2}\sqrt{\frac{9}{8}+\frac{\sqrt{13}}{8}}\right]\right\}\right\}$$

```
%//N          ← 这时数值解可能更有意义.
{{x → 1.40492}, {x → 0.165873}, {x → 2.83487}, {x→ -1.26407}}
```

7.22 求和并化简：$\dfrac{\cos x}{1+\sin x}+\tan x$.

解 ☞

```
Together[Cos[x]/(1 + Sin[x]) + Tan[x], Tring → True]
```

$$\frac{1}{\left(\text{Cos}\left[\frac{x}{2}\right]-\text{Sin}\left[\frac{x}{2}\right]\right)\left(\text{Cos}\left[\frac{x}{2}\right]+\text{Sin}\left[\frac{x}{2}\right]\right)}$$

```
TrigReduce[ % ]
Sec[x]
```

> 有时候可能必须使用两条或多条命令，才能完全简化表达式.

7.23 组合并化简：$\dfrac{\sinh x}{\cosh x-\sinh x}+\dfrac{\cosh x}{\cosh x+\sinh x}$.

解 ☞

```
Together[Sinh[x]/(Cosh[x] - Sinh[x]) + Cosh[x]/(Cosh[x] + Sinh[x]), Trig → True]
Cosh[2x]
```

7.24 构造 $\sin nx$ 与 $\cos nx$ 对 $n=2, 3, 4, 5$ 的倍角公式表.

解 ☞

```
trigtable = Table[{n, TrigExpand[Sin[n x]], TrigExpand[Cos[n x]]}, {n, 2, 5}];
TableForm[trigtable, TableHeadings →{None, {"n", "sin nx", "cos nx"}}]
```

n	sin nx	cos nx
2	2Cos[x]Sin[x]	$\text{Cos}[x]^2-\text{Sin}[x]^2$
3	$3\text{Cos}[x]^2\text{Sin}[x]-\text{Sin}[x]^3$	$\text{Cos}[x]^3-3\text{Cos}[x]\text{Sin}[x]^2$
4	$4\text{Cos}[x]^3\text{Sin}[x]-4\text{Cos}[x]\text{Sin}[x]^3$	$\text{Cos}[x]^4-6\text{Cos}[x]^2\text{Sin}[x]^2+\text{Sin}[x]^4$
5	$5\text{Cos}[x]^4\text{Sin}[x]-10\text{Cos}[x]^2\text{Sin}[x]^3+\text{Sin}[x]^5$	$\text{Cos}[x]^5-10\text{Cos}[x]^3\text{Sin}[x]^2+5\text{Cos}[x]\text{Sin}[x]^4$

7.25 构造 $\sin^n x$ 与 $\cos^n x$ 对 $n=2, 3, 4, 5$ 的线性三角公式表.

解 ☞

```
trigtable = Table[{n, TrigReduce[Sin[x]^n], TrigReduce[Cos[x]^n]}, {n, 2, 5}];
TableForm[trigtable, TableHeadings →{None, {"n", "sin^n x","cos^n x"}}]
```

n	$\sin^n x$	$\cos^n x$
2	$\frac{1}{2}(1-\text{Cos}[2x])$	$\frac{1}{2}(1+\text{Cos}[2x])$
3	$\frac{1}{4}(3\text{Sin}[x]-\text{Sin}[3x])$	$\frac{1}{4}(3\text{Cos}[x]+\text{Cos}[3x])$
4	$\frac{1}{8}(3-4\text{Cos}[2x]+\text{Cos}[4x])$	$\frac{1}{8}(3+4\text{Cos}[2x]+\text{Cos}[4x])$
5	$\frac{1}{16}(10\text{Sin}[x]-5\text{Sin}[3x]+\text{Sin}[5x])$	$\frac{1}{16}(10\text{Cos}[x]+5\text{Cos}[3x]+\text{Cos}[5x])$

7.26 把 e^{x+y} 表示成双曲函数的形式,并展开.

解 ☞

```
ExpToTrig[E^(x+y)]
Cosh[x + y] + Sinh[x + y]
TrigExpand[%]
Cosh[x]Cosh[y] + Cosh[y]Sinh[x] + Cosh[x]Sinh[y] + Sinh[x]Sinh[y]
```

7.27 把 $\sinh^{-1}x$ 与 $\tan^{-1}x$ 表示成指数形式.

解 ☞

```
TrigToExp[ArcSinh[x]]
```

$\text{Log}[x+\sqrt{1+x^2}]$

```
TrigToExp[ArcTanh[x]]
```

$-\frac{1}{2}\text{Log}[1-x]+\frac{1}{2}\text{Log}[1+x]$

7.4 化简的技巧

对于特定的代数或三角表达式,可以表示成许多不同的形式,而且不同的人对于"简单"形式的看法可能不一样.例如,在处理有理函数的时候,$(x+3)^2$ 可能比 x^2+6x+9 更精简,但在处理多项式时,所要求的最简形式就应当是后者.

正如你在本章中所见到的那样，Mathematica 提供了许多命令控制表达式的形式．在实际使用过程中，你就会逐步学会如何使用这些命令改变表达式的形式，以满足自己的需要．

作为迈向简化的一个步骤，Mathematica 提供了两条命令，以化简复杂结构．

- **Simplify[表达式]** 对表达式进行一系列的变换，并返回它所找到的最简形式．
- **FullSimplify[表达式]** 对表达式进行包括初等与特殊函数在内的更大范围的变换，并返回它所找到的最简形式．

Simplify 会尝试利用展开、因式分解以及其它标准数学变换简化表达式的复杂性．由于它的一般性，Simplify 一般要比更直接的命令慢很多．FullSimplify 得到的结果至少不会比 Simplify 的结果复杂，但花费的时间可能更长．

可以用选项 **TimeConstraint** 指定一个时间限度（以秒为单位）．对于 **Simplify** 默认值为 **TimeConstraint →300**，对 **FullSimplify**，**TimeConstraint →Infinity**．对于这两条命令，**Trig →True** 都是进行三角运算时的默认值．

例 30　首先，生成一个很长的代数表达式．

messyexpression = Expand$\left[\left(\frac{1}{x+1}+\frac{1}{x+2}+\frac{1}{x+3}\right)^5\right]$

$$\frac{1}{(1+x)^5}+\frac{1}{(2+x)^5}+\frac{5}{(1+x)\,(2+x)^4}+\frac{10}{(1+x)^2(2+x)^3}+\frac{10}{(1+x)^3(2-x)^2}+$$
$$\frac{5}{(1+x)^4(2+x)}+\frac{1}{(3+x)^5}+\frac{5}{(1+x)(3+x)^4}+\frac{5}{(2+x)\,(3+x)^4}+\frac{10}{(1+x)^2(3+x)^3}+$$
$$\frac{10}{(2+x)^2(3+x)^3}+\frac{20}{(1+x)\,(2+x)\,(3+x)^3}+\frac{10}{(1+x)^3(3+x)^2}+\frac{10}{(2+x)^3(3+x)^2}+$$
$$\frac{30}{(1+x)\,(2+x)^2(3+x)^2}+\frac{30}{(1+x)^2(2+x)\,(3+x)^2}+\frac{5}{(1+x)^4(3+x)}+\frac{5}{(2+x)^4(3+x)}+$$
$$\frac{20}{(1+x)\,(2+x)^3(3+x)}+\frac{30}{(1+x)^2(2+x)^2(3+x)}+\frac{20}{(1+x)^3(2+x)\,(3+x)}$$

现在化简这个表达式．当然，Mathematica 并不会记住 messyexpression 是从哪儿来的．

Simplify[messyexpression]

$$\frac{(11+12x+3x^2)^5}{(1+x)^5(2+x)^5+(3+x)^5}$$

例 31　**messytrigexpression = Expand[(Tan[x]^2 + Sin[x]^2 + Cos[x]^2)^5]**

Cos[x]^{10} + 5Cos[x]^{6}Sin[x]^2 + 5Cos[x]^{8}Sin[x]^2 + 10Cos[x]^{2}Sin[x]^4 +
20Cos[x]^{4}Sin[x]^4 + 10Cos[x]^{6}Sin[x]^4 + 30Sin[x]^6 + 30Cos[x]^{2}Sin[x]^6 +
10Cos[x]^{4}Sin[x]^6 + 20Sin[x]^8 + 5Cos[x]^{2}Sin[x]^8 + Sin[x]^{10} + 10Sin[x]^{4}Tan[x]^2 +
30Sin[x]^{6}Tan[x]^2 + 5Sin[x]^{8}Tan[x]^2 + 20Sin[x]^{4}Tan[x]^4 + 10Sin[x]^{6}Tan[x]^4 +
5Sin[x]^{2}Tan[x]^6 + 10Sin[x]^{4}Tan[x]^6 + 5Sin[x]^{2}Tan[x]^8 + Tan[x]^{10}

Simplify[messytrigexpression]

Sec[x]^{10}

第八章 微分运算

8.1 极限的计算

函数的极限是微分的基石.对于复杂函数,极限的计算相当困难,需要各种专门的计算方法.Mathematica内置了完成这一任务的程序,它总是尽力确定极限的准确值.

■ **Limit[f[x], x →a]** 计算$\lim\limits_{x\to a}f(x)$的值.

例1 希望计算$\lim\limits_{x\to 2}\frac{x^5-32}{x^3-8}$的值.由于当$x\to 2$时,分子与分母的值都趋于零,因此极限值并不是一目了然的.

Limit[$\frac{x^5-32}{x^3-8}$, x →2]

$\frac{20}{3}$

可以利用 **Direction** 选项确定左右极限的计算.

- **Direction →1** 使得极限作为左极限来计算,即x从左面趋近于a.
- **Direction → - 1** 使得极限作为右极限来计算,即x从右面趋近于a.

Limit 的默认值为 **Direction →Automatic**,除了在∞点外,它的值为 Direction → - 1.因此对于不连续函数,如果没有给出 Direction 选项,Mathematica 给出的极限值可能不正确.

例2 考虑极限$\lim\limits_{x\to 0}\frac{|x|}{x}$.

Lim[$\frac{\text{Abs[x]}}{\text{x}}$, x →0]

1

由于没有指定方向选项,因此在默认情况中只计算右极限.为了对极限进行完全地分析,我们也计算它的左极限.

Lim[$\frac{\text{Abs[x]}}{\text{x}}$, x →0, Direction →1]

- 1

由于左右极限不同,所以上述极限不存在.

Mathematica 也可以计算无穷极限以及在∞处的极限.

例3 **Limit[1/x, x →0, Direction → - 1]**

∞

Limit[1/x, x →0, Direction →1]

- ∞

Limit[$\frac{2x^2+3x+4}{x^2+1}$, x →∞]

2

在下面例子中函数演示了另外一种完全不同的行为.当 $x \to 0$ 时,函数要来回振动无穷次.*Mathematica* 返回的极限为区间对象 Interval[{min, max}],表示值的范围介于 min 与 max 之间.

例 4

```
Limit[Sin[1/x], x→0]
Interval[{-1, 1}]
Limit[Tan[1/x], x→0]
Interval[{-∞, ∞}]
```

习题解答

8.1 计算极限 $\lim\limits_{x\to 0}\dfrac{2^x+x-1}{3x}$.

解 ☞

$\text{Limit}[\dfrac{2^x+x-1}{3x},\ x\to 0]$

$\dfrac{1}{3}(1+\text{Log}[2])$

8.2 计算极限 $\lim\limits_{x\to 0}\dfrac{\tan x - x}{x^3}$.

解 ☞

$\text{Limit}[\dfrac{\text{Tan}[x]-x}{x^3},\ x\to 0]$

$\dfrac{1}{3}$

8.3 计算极限 $\lim\limits_{x\to 0}(1+\sin x)^{\cot 2x}$.

解 ☞

$\text{Limit}[(1+\text{Sin}[x])^{\text{Cot}[2x]},\ x\to 0]$

$\sqrt{e}$

8.4 计算极限 $\lim\limits_{x\to\infty}(e^x+x)^{1/x}$ 与 $\lim\limits_{x\to -\infty}(e^x+x)^{1/x}$.

解 ☞

$\text{Limit}[(\text{Exp}[x]+x)^{1/x},\ x\to\infty]$

e

$\text{Limit}[(\text{Exp}[x]+x)^{1/x},\ x\to -\infty]$

1

8.5 计算极限 $\lim\limits_{x\to 1}(2-x)^{\tan(\frac{\pi}{2}x)}$.

解 ☞

$\text{Limit}[(2-x)^{\text{Tan}[\frac{\pi}{2}x]},\ x\to 1]$

$e^{2/\pi}$

8.6 如果每年支付利息 n 次，年利率为 r，那么 p 美元在 t 年后变为 $p\left(1+\frac{r}{n}\right)^{nt}$ 美元，假设如果连续计息($n \to \infty$)，那么 t 年后钱数为多少？

解 ☞

```
Limit[p(1 + r/n)^(n t), n → ∞]
e^(r t) p
```

8.7 函数的导数定义为极限 $\lim_{h\to 0}\frac{f(x+h)-f(x)}{h}$. 利用这个定义计算 $f(x)=\ln x + x^5 + \sin x$ 的导数.

解 ☞

```
f[x_] = Log[x] + x^5 + Sin[x];
Limit[(f(x + h) - f(x))/h, h → 0]
1/x + 5x^4 + Cos[x]
```

8.8 函数的 2 阶导数可以用 $\lim_{h\to 0}\frac{f(x+h)-2f(x)+f(x-h)}{h^2}$ 极限计算得到. 利用这个极限计算 $f(x)=\ln x + x^5 + \sin x$ 的 2 阶导数.

解 ☞

```
f[x_] = Log[x] + x^5 + Sin[x];
Limit[(f[x + h] - 2f[x] + f[x - h])/h^2, h → 0]
-1/x^2 + 20x^3 - Sin[x]
```

8.2 导数的计算

在 Mathematica 中有几种计算导数的方法. 每种方法都有其优势与不足，因此对于特定的问题，必须进行适当的选择.

■ 如果 **f[x]** 表示一个函数，那么它的导数表示为 **f′[x]**. 高阶导数用 **f″[x]**，**f‴[x]**，…等表示.

例 5

```
f[x_] = x^5 + x^4 + x^3 + x^2 + x + 1;
f'[x]
1 + 2x + 3x^2 + 4x^3 + 5x^4
f''[x]
2 + 6x + 12x^2 + 20x^3
f'''[x]
6 + 24x + 60x^2
```

撇号也可以作用到内置函数上，见下面例子所示. 如果不给出参数，Mathematica 就返回一个纯粹函数，表示所要求的导数(纯粹函数的讨论见附录).

例 6

```
Sqrt'
```

$\frac{1}{2\sqrt{\#1}}$&

Sqrt′[x]

$\frac{1}{2\sqrt{x}}$

Sqrt″

$-\frac{1}{2(2\#1^{3/2})}$&

Sqrt″[x]

$-\frac{1}{4x^{3/2}}$

■ D[f[x], x] 返回 f 相应于变量 x 的导数.

■ D[f[x], {x, n}] 返回 f 相应于变量 x 的 n 阶导数.

例 7 D[x^5 + x^4 + x^3 + x^2 + x + 1, x]

1 + 2x + $3x^2$ + $4x^3$ + $5x^4$

D[x^5 + x^4 + x^3 + x^2 + x + 1, {x, 2}]

2 + 6x + $12x^2$ + $20x^3$

D[x^5 + x^4 + x^3 + x^2 + x + 1, {x, 3}]

6 + 24x + $60x^2$

■ $\partial_{\square}$,可以在 BasicInput 模板中找到这个符号,它等价于 D 命令,∂_x 返回相应于 x 的导数.n 阶导数表示为$\partial_{\{x, n\}}$.

例 8 ∂_x(x^5 + x^4 + x^3 + x^2 + x + 1)

1 + 2x + $3x^2$ + $4x^3$ + $5x^4$

$\partial_{\{x,2\}}$(x^5 + x^4 + x^3 + x^2 + x + 1)

2 + 6x + $12x^2$ + $20x^3$

$\partial_{\{x,3\}}$(x^5 + x^4 + x^3 + x^2 + x + 1)

6 + 24x + $60x^2$

■ Derivative[n] 是一个泛函算子,它作用到一个函数上,得到一个新的函数,即函数的 n 阶导函数.Derivative[n][f]用纯粹函数的形式给出 f 的 n 阶导数,而 Derivative[n][f][x]计算 f 在 x 点的 n 阶导数.

有一些事实值得记住,那就是在 Mathematica 的内部,f′要被转化为 Derivative[1][f].因此 f′[x]成为 Derivative[1][f][x].f″,f‴等等的处理类似.

例 9 f[x_] = x^5 + x^4 + x^3 + x^2 + x + 1;

Derivative[1][f]

1 + 2#1 + 3#1^2 + 4#1^3 + 5#1^4 &　　←Mathematica 返回一个纯粹函数表示 f 的导数.在附录中对纯粹函数进行了讨论.

Derivative[1][f][x]

1 + 2x + $3x^2$ + $4x^3$ + $5x^4$

要计算导数在特定点的值,可以采用几种方法,具体使用哪种方法,要根据导数的计算命令确定.下面的例子演示了最常用的方法.

例 10

```
f[x_] = (x^2 - x + 1)^5;
f''[1]
30
D[f[x], {x, 2}] /. x → 1
30
∂{x,2} f[x] /. x → 1
30
g := Derivative[2][f]
g[1]
30
f[x_] = x^3;
g[1]
6
```

← 对本例前三部分中的每一部分而言,都是先计算 2 阶导数,然后再用 1 代替 x.

← 这里我们定义了一个新函数 g,表示 f 的 2 阶导数.如果 f 发生了改变,g 就变为新函数的 2 阶导数.注意这里使用了 :=.如果需要 g 反应 f 的改变,这一点非常重要.

Mathematica 通过"记住"各种法则来计算函数的复合、和、差、积与商以及它们的组合的导数.如果没有给出函数的定义,那么就可以看出这些法则的内容.

例 11

```
Clear[f, g]
D[f[x] + g[x], x]
f'[x] + g'[x]                      ← 和的导数等于各项导数的和.
D[f[x] g[x], x]
g[x] f'[x] + f[x] g'[x]            ← 这就是熟悉的乘积法则.
D[f[x]/g[x], x]//Together
(g[x]f'[x] - f[x]g'(x)) / g[x]^2   ← 商法则.
D[f[g[x]], x]
f'[g[x]]g'[x]                      ← 链式法则.
```

我们可以通过观察图形,利用 Mathematica 发现一些基本理论.由罗尔定理可知,在某些条件下,存在一点,该点导数值为零:

> 令 f 为有限闭区间 $[a, b]$ 上的连续函数,并在开区间 (a, b) 上可微,假设 $f(a) = f(b) = 0$.那么存在一个数 c,介于 a 与 b 之间,使得 $f'(c) = 0$.

也就是说,如果光滑函数在 a 与 b 两点同时等于零,那么在这两点之间必存在一点,它的导数等于零.

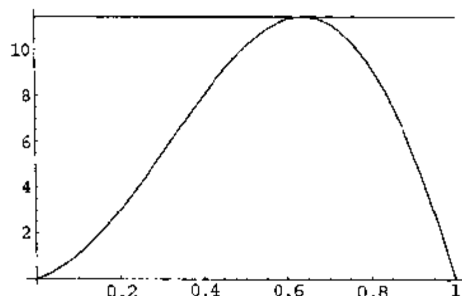
例 12 证明函数 $f(x) = (x^3 + 2x^2 + 15x + 2)\sin \pi x$ 在区间 $[0, 1]$ 上满足罗尔定理,并求出定理中所声称的点 c.

由于 f 是多项式与正弦三角函数的乘积,所以 f 处处连续而且可微.

```
f[x_] = (x^3 + 2x^2 + 15x + 2)Sin[πx];
f[0]
0
f[1]
```

```
0
FindRoot[f'[c] == 0, {c, 0.5}]
{c → 0.640241}
Plot[{f[x], f[.640241]}, {x, 0, 1}];
```

← 我们用 0.5 作为初始估计值,因为它是 0 与 1 的中点.



中值定理与罗尔定理类似,但它不要求 f 在区间的两个端点上取零值:

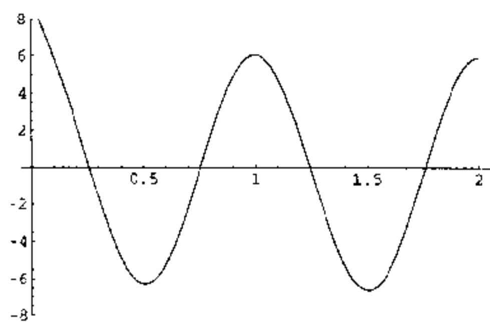
令 f 为有限闭区间 $[a, b]$ 上的连续函数,并在开区间 (a, b) 上可微. 那么存在一个数 c,介于 a 与 b 之间,使得 $f(b)-f(a)=f'(c)(b-a)$.

如果定理的结论写作 $\frac{f(b)-f(a)}{b-a}=f'(c)$ 的形式,我们可见中值定理保证了在 a 与 b 之间存在一点 c,在 $(c, f(c))$ 点的切线平行于连接曲线两个端点的线段.

注意 罗尔定理与中值定理都只保证至少存在一个数 c,实际上对于具体的问题,可能存在满足条件的多个数.

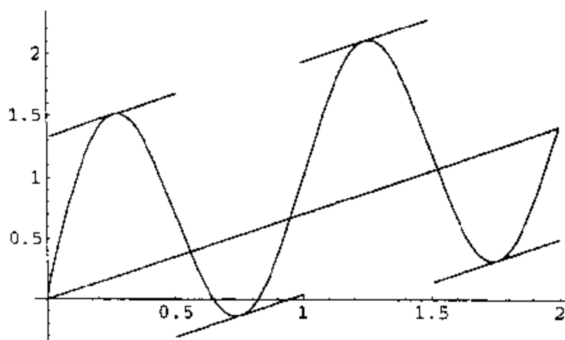
例 13 对于函数 $f(x)=\sqrt{x}+\sin 2\pi x$,求出使得中值定理在区间 $[0, 2]$ 上成立的值 c.

```
f[x_] = √x + Sin[2πx];
a = 0; b = 2;
m = (f[b] - f[a])/(b - a);
Plot[f'[x] - m, {x, 0, 2}, PlotRange → {-8, 8}]
```



为了确定 c 的位置,我们需要估计函数 $f'(x)-m$ 的零点。由图可知它有四个零点,分别近似为 0.3,0.7,1.3,1.7.

```
FindRoot[f'[c] == m, {c, .3}]
FindRoot[f'[c] == m, {c, .7}]
FindRoot [ f' [ c ] == m, {c, 1.3}]
FindRoot [ f' [ c ] == m, {c, 1.7}]
{c → 0.257071}
{c → 0.753319}
{c → 1.24344}
```



由图可见,切线平行于曲线端点的连线

```
{c → 1.75836}
```

习题解答

8.9　计算 $\tan x$ 的 3 阶导数.

解 ☞

```
Tan'''[x] //Expand        ← 对于 Mathematica 版本 3,需要插入//Expand 在版本 4 中会自动展开的.
2 Sec[x]^4 + 4 Sec[x]^2 Tan[x]^2
```

8.10　计算 $f(x)=e^{x^2}$ 在 $x=0$ 点的前 10 阶导数的值,把结果以表格的形式列出来.

解 ☞

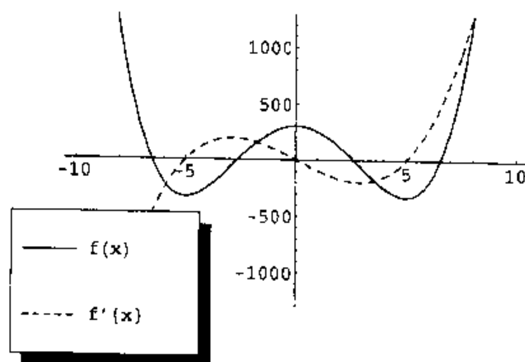
```
f[x_] = Exp[x^2];
derivtable = Table[{n, D[f[x], {x, n}]/. x→0}, {n, 1, 10}];
TableForm[derivtable,
        TableAlignments →Center,
        TableDirections → Row,
        TableSpacing → 2,
        TableHeadings →{None, {"n", "f^(n)(0) "}}]
```

n	1	2	3	4	5	6	7	8	9	10
$f^{(n)}(0)$	0	2	0	12	0	120	0	1680	0	30240

8.11　在同一坐标系中画出 $f(x)=x^4-50x^2+300$ 及其导数在 $-10\leqslant x\leqslant 10$ 上的图形.

解 ☞

```
<<Graphics`Legend`
f[x_] = x^4 - 50x^2 + 300;
Plot[{f[x], f'[x]}, {x, -10, 10},
    PlotStyle →{GrayLevel[0], Dashing[{.015}]},
    PlotLegend →{"f[x]", "f'[x]"}];
```

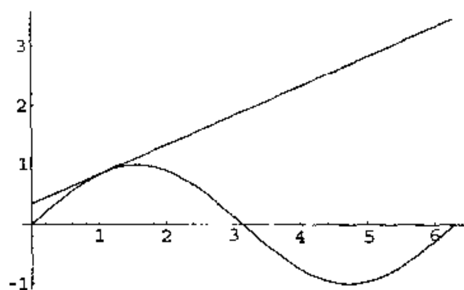


注意观察当 $f'(x)=0$ 时 $f(x)$ 具有相对(局部)极大值或极小值.

8.12　给定函数 $f(x)$,其图形为 C,C 在点 a 的切线斜率为 $f'(a)$.令 $f(x)=\sin x$,画出函数图形及其在 $a=\pi/3$ 处的切线.

解 ☞

```
f[x_] = Sin[x];
a = π/3;
l[x_] = f[a] + f′[a](x - a);
Plot[{f[x], l[x]}, {x, 0, 2π}];
```



注意过点(x_1, y_1)，斜率为 m 的直线方程为

$$y - y_1 = m(x - x_1)$$

或者 $$y = y_1 + m(x - x_1).$$

这里 $x_1 = a$，$y_1 = f(a)$，$m = f'(a)$，因此

$$y = f(a) + f'(a)(x - a)$$

8.13 构造一个动画，演示曲线 $y = \sin x$，$0 \leqslant x \leqslant 2\pi$ 的切线沿曲线运动的情形.

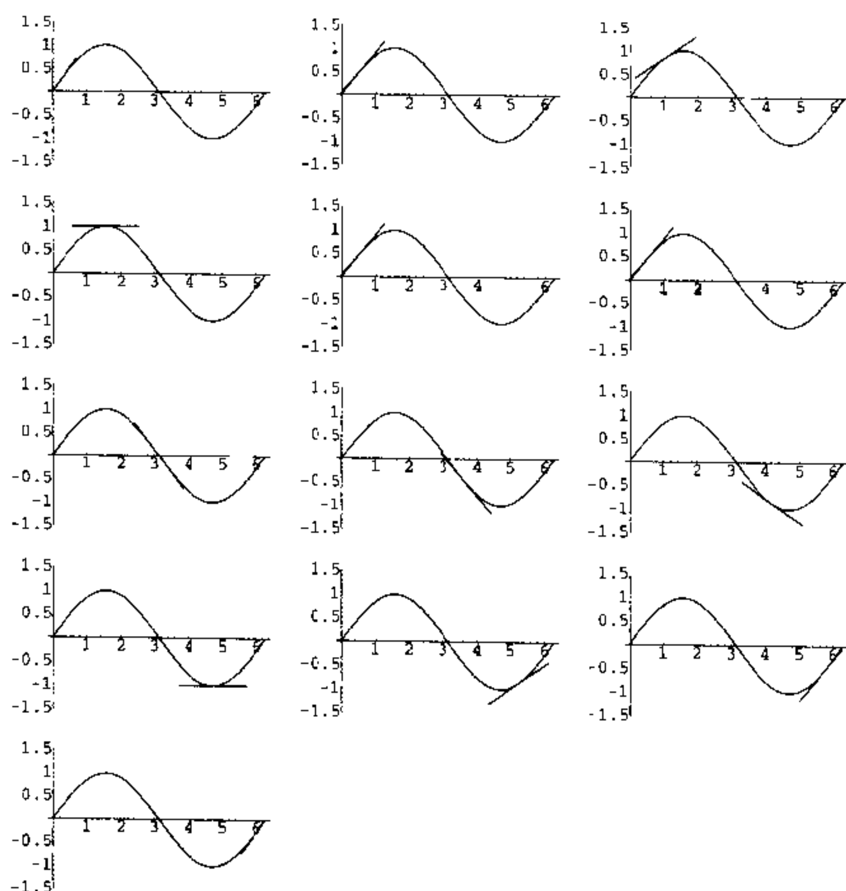
解 ☞

在设计图形的切线时，我们希望它具有常数长度，如 2 个单位长，而且切点就是它的中点. 为此，我们构造函数 l[x]，它在 $a - 1/\sqrt{1+m^2}$ 与 $a + 1/\sqrt{1+m^2}$（m 表示切线的斜率）之间定义了在 x 点的切线.

然而这时有一个小问题. 当我们要绘制 l[x]在区间 $0 \leqslant x \leqslant 2\pi$ 上的图形时，由于 l[x]只是定义在这个区间的子集上，所以 Mathematica 会给出出错信息. 可以用命令 Off[Plot::plnr]关掉这条出错信息的显示.

```
f[x_] = Sin[x];
m := f′[a]
l[x_] = f[a] + m(x - a)/; a - 1/√(1 + m²) ≤ x ≤ a + 1/√(1 + m²)
Do[Plot[{f[x], l[x]}, {x, 0, 2π},
    PlotRange → {-1.5, 1.5}], {a, 0, 2π, π/6}]
Plot::plnr :              ← 这条消息要重复很多次.
l[x] is not a machine - size real number at x = 1.04500937601917009`.
(l[x]在 x = 1.04500937601917009`不是机器认可的实数.)
Off[Plot::plnr]           ← 去掉出错信息.
Do[Plot[{f[x], l[x]}, {x, 0, 2π},
    PlotRange → {-1.5, 1.5}], {a, 0, 2π, π/6}]
On[Plot::plnr]            ← 恢复显示这条出错信息的能力.
```

选择图形单元括号，然后进入 Cell 菜单，点击 **Cell ⇒Animate Selected Graphics** 就可以查看动画.



8.14 对于区间 [0, 4]上的函数 $f(x)=4x + 39x^2 - 46x^3 + 17x^4 - 2x^5$，求出使罗尔定理成立的值 c.

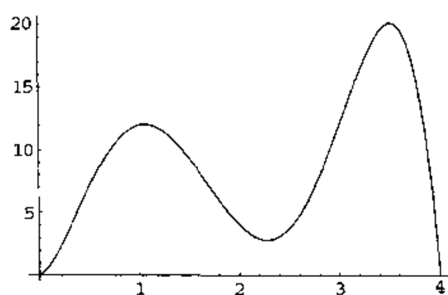
解 ☞

由于 $f(x)$ 为多项式，所以它处处连续而且可微. 首先验证 $f(0)=f(x)=0$.

```
f[x_] = 4x + 39x^2 - 46x^3 + 17x^4 - 2x^5;
f[0]
0
f[4]
0
```

现在看一下 $f'(c)=0$ 在哪儿成立. 由于 f' 为多项式，所以可以用 **NSolve** 命令求解.

```
NSolve[f'[c]==0]
{{c → -0.0472411}, {c → 1.05962},
{c → 2.27466}, {c → 3.51296}}
```



因此在 0 与 4 之间存在三个 c 值.（罗尔定理只保证至少存在一个.）函数的图形验证了这一点.

```
Plot[f[x], {x, 0, 4}]
```

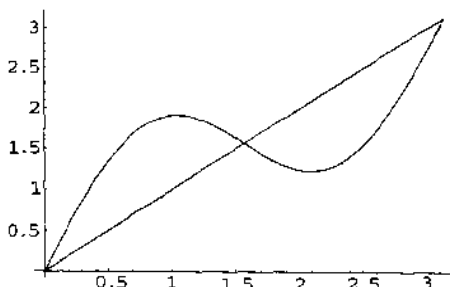
8.15 对于区间[0, π] 上的函数 $f(x)=x + \sin 2x$，验证中值定理成立.

解 ☞

$f(x)$ 是处处连续而且可微的. 定义 $a=0$，$b=2\pi$，并求解关于 c 的方程 $f(b)$

$-f(a)=f'(c)(b-a)$. 为了给出 c 的近似值，我们查看一下函数的图形以及曲线端点的连线.

```
f[x_] = x + Sin[2x];
a = 0; b = π;
m = (f[b] - f[a])/(b - a);        ← 连接端点的割线的斜率.
l[x_] = f[a] + m(x - a);          ← 表示割线的函数.
Plot[{f[x], l[x]}, {x, a, b}];
```



由图可见，在 $x=1$ 与 $x=2.5$ 附近的切线平行于割线.

```
FindRoot[f[b] - f[a] == f'[c](b - a), {c, 1}]
FindRoot[f[b] - f[a] == f'[c](b - a), {c, 2.5}]
{c → 0.785398}
{c → 2.35619}
```

这两个值都位于 0 与 π 之间.

8.3 最大值与最小值

我们称函数 f 在区间 I 中 c 点达到绝对(全局)最大值，是指对 I 中所有点 x，$f(x)\leqslant f(c)$ 成立. 也就是说，$f(c)$ 是 $f(x)$ 在 I 上的最大值. 绝对最小值的定义类似(只是不等式反号). 微分学有一个最重要的应用就是优化问题的求解，即在某种限制下求出函数的最大值与最小值.

并不是所有的函数都具有绝对最大值和最小值. 然而最值定理给出的条件保证了它的存在性：

如果 f 为有界闭区间上的连续函数，那么 f 在这个区间既具有绝对最大值，也具有绝对最小值.

函数 f 的临界点就是数 c 满足 $f'(c)=0$ 或者 $f'(c)$ 不存在. 在本书中我们只考虑可微函数，因此临界点也就是导数值等于零的点.

可以证明如果函数在有界闭区间 $[a,\ b]$ 上连续，那么它的绝对最大值和最小值就在临界点或者区间端点上达到. 由此我们可以利用 Mathematica 帮助我们计算函数的最大值和最小值.

例 14 计算函数 $f(x)=x^4-4x^3+2x^2+4x+2$ 在区间 $[0,4]$ 上的绝对最大值和最小值. 首先求出它的临界点.

```
f[x_] = x^4 - 4x^3 + 2x^2 + 4x + 2;
Solve[f'[x] == 0]
{{x→1}, {x→1 - √2}, {x→1 + √2}}
```

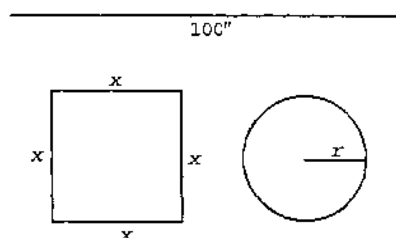
这三个数中只有两个在区间 [0, 4] 中.我们计算函数在这些值以及区间端点上的值.

```
c1 = 0; c2 = 1; c3 = 1 + √2; c4 = 4;
pointstocheck = {{c1, f[c1]}, {c2, f[c2]}, {c3, f[c3]}, {c4, f[c4]}}//Expand;
TableForm[poinstocheck, TableHeading →{None, {"x", "f[x]"}}]
```

x	f[x]
0	2
1	5
$1+\sqrt{2}$	1
4	50

所以 f 在该区间上的绝对最大值为 50,绝对最小值为 1.

例 15 一根线长 100 英寸,要用它构成一个正方形和一个圆形.请问如何分配,才能使它所构成的图形面积和 (a) 最大;(b) 最小.



两个形状的组合面积为 $A(x)=x^2+\pi r^2$.圆的周长为 $2\pi r$,正方形周长为 $4x$,而线长为 100,所以 $4x+2\pi r=100$,并且 $0 \leqslant x \leqslant 25$.

```
Solve[4 x + 2 π r == 100, r]
```

$$\left\{\left\{r \to -\frac{2(-25+x)}{\pi}\right\}\right\}$$

```
a[x_] = x^2 + π r^2/. r → -2(-25 + x)/π        ←用 x 代替 r.
```

$$\frac{4(-25+x)^2}{\pi}+x^2$$

```
Solve[a'[x] == 0]        ← 求出临界点.
```

$$\left\{\left\{x \to \frac{100}{4+\pi}\right\}\right\}$$

```
x1 = 0;
x2 = 100/(4 + π);
x3 = 25;
pointstocheck = {{x1, a[x1], N[a[x1]]}, {x2, a[x2], N[a[x2]]},
        {x3, a[x3], N[a[x3]]}}//Together;
TableForm[pointstocheck, TableAlignments → Center,
        TableHeadings →{None, "x", "a[x]", "N[a[x]]"}}]
```

计算 $a(x)$ 在这三点的值.为了便于比较,把它们的近似值放在一个表格中.

x	a[x]	N[a[x]]
0	$\frac{2500}{\pi}$	795.775
$\frac{100}{4+\pi}$	$\frac{2500}{4+\pi}$	350.062
25	625	625.

由上可见,最大面积出现在 $x=0$(即所有的线都用来构成圆).最小面积是在正方形边长为$\frac{100}{4+\pi}$时达到(即从线中剪下$\frac{400}{4+\pi}$构成正方形).

为了进一步验证 $x=\frac{100}{4+\pi}$确实给出了最小值,我们进行2阶导数检测.

```
Sign[a''[100/(4 + π)]]
1
```

> 由于在临界点2阶导数的值为正的,所以在 100/(4+π)达到相对极小值.而这是惟一的相对极小值,所以它一定是绝对最小值.

称函数 f 在 c 点达到相对(或局部)极大值是指存在包含 c 点的开区间 I,使得对于 I 中所有点 x, $f(x)\leqslant f(c)$ 成立.也就是说存在一个包含 c 的开区间使得 $f(c)$ 为 f 在这个区间上的最大值.相对极小值的定义类似.

与绝对最大值(最小值)不同,函数可能具有几个相对极大值(极小值).如果希望知道它们所有的近似位置,那么可以用 Mathematica 命令 **FindMinimum** 有效而方便地计算出来.

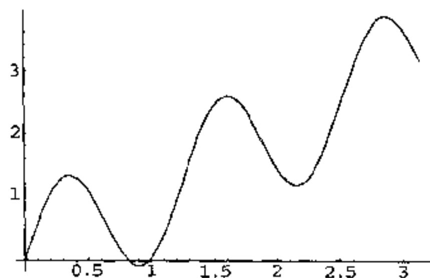
■ **FindMinimum[f[x], {x, x0}]** 求出 $f(x)$靠近 $x0$ 点的相对极小值.

FindMinimum 利用修正的最速下降法计算函数的相对极小值.可以用选项 **Method → Newton** 或 **Method → QuasiNewton** 改变所使用的方法.

与 FindRoot 一样,如果希望得到更高的准确度,可以设置 AccuracyGoal 和 WorkingPrecision 选项.另外,可以设置 PrecisionGoal 以确定函数在极小值点值的精度.(所谓精度,就是答案中的有效数字位数;而准确度则指的是小数点右边的有效数字位数.)

虽然 Mathematica 并没有提供计算相对极大值的命令,但 **- FindMinimum[- f[x], {x, x0}]**就可以做到这一点(在答案中会出现额外的负号,但可以忽略).

例 16 函数 $f(x)=x+\sin 5x$ 在区间 $[0,\pi]$ 中有三个相对极大值点,两个相对极小值点.只要看一下函数的图形,就能知道它们的近似位置.



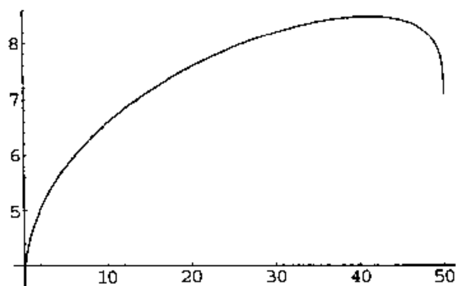

```
f[x_] = x + Sin[5x];
Plot[f[x], {x, 0, π}];
FindMinimum[f[x], {x, 1}]
{-0.0775897, {x → 0.902206}}        ← 首先显示的是函数值,然后是 x 的值.
FindMinimum[f[x], {x, 2}]
{1.17905, {x → 2.15884}}
-FindMinimum[-f[x], {x, 0.4}]
{1.33423, {-(x → 0.354431)}}        ← 可以忽略这个负号(-).
-FindMinimum[-f[x], {x, 1.6}]
{2.59086, {-(x → 1.61107)}}
-FindMinimum[-f[x], {x, 2.8}]
{3.8475, {-(x → 2.8677)}}
```

因此相对极大值点为 (0.354431, 1.33423), (1.61107, 2.59086) 与 (2.8677, 3.8475). 极小值点为 (0.902206, - 0.0775897) 与 (2.15884, 1.17905).

习题解答

8.16 求出两个和为 50 的正数,使得第一个数的平方根加上第二个数的立方根尽可能大.

解 ☞




```
y = 50 - x;
f[x_] = √x + ∛y;
NSolve[f'[x] == 0]
{{x → 41.1553}}
-FindMinimum[-f[x], {x, 40}]
```
← 另外一种方法.

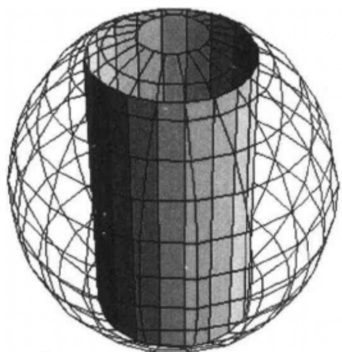
```
{8.48329, {-(x → 41.1553)}}
y/.x →41.1553
8.8447
```

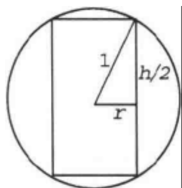
因此两个数为 $x = 41.1553$ 与 $y = 8.8447$. 最大和为 8.48329.

```
Plot[f[x], {x, 0, 50}]
```

8.17 一个圆柱被单位球面所截. (a) 求出最大可能的体积. (b) 求出最大可能的表面面积.

解 ☞





(a) 考虑问题的二维透视图. 记被截圆柱的半径与高分别为 r 与 h. 那么被截圆柱的体积为 $V = \pi r^2 h$, 由勾股定理可知 $r^2 + \left(\frac{h}{2}\right)^2 = 1$. 由此容易知道(即使不用 Mathematica) $r^2 = 1 - \left(\frac{h}{2}\right)^2$, 所以体积为 h 的函数, 表示为 $V(h) = \pi\left[1 - \left(\frac{h}{2}\right)^2\right]h$.

```
v[h_] = π(1 - (h/2)^2)
Solve[v'[h] == 0, h]
{{h → -2/√3}, {h → 2/√3}}
vmax = v[2/√3]
```
← 显然只有 h 的正值是适宜的.

$\frac{4\pi}{3\sqrt{3}}$

```
Sign[v''[2/√3]]
```

-1

由于在临界点的 2 阶导数为负数，所以 $2/\sqrt{3}$ 点为相对极大值点。而这是惟一的相对极值点，所以它必是绝对最大值点.

(b) 圆柱的表面积(包含顶与底)为 $S=2\pi rh+2\pi r^2$. 与 (a) 中一样，仍有 $r^2+\left(\frac{h}{2}\right)^2=1$，但由于在公式 S 中同时出现了 r 与 r^2，所以用 r 表示 h 更容易处理.

```
Solve[r^2 + (h/2)^2 == 1, h]
```

$\{\{h\to -2\sqrt{1-r^2}\},\{h\to 2\sqrt{1-r^2}\}\}$

现在把 h 的正值代入到关于 S 的公式中：

```
s[r_] = 2π r h + 2 π r^2/. h →2√(1 - r^2)
```

$2\pi r^2+4\pi r\sqrt{1-r^2}$

```
s'[r]
```

$4\pi r-\frac{4\pi r^2}{\sqrt{1-r^2}}+4\pi\sqrt{1-r^2}$

这时求解 r 的临界点：

```
Solve[s'[r] == 0, r]//Simplify
```

$\left\{\left\{r\to-\sqrt{\frac{1}{10}(5-\sqrt{5})}\right\},\left\{r\to\sqrt{\frac{1}{10}(5+\sqrt{5})}\right\}\right\}$

只有 r 的正值是有用的，我们就用它计算最大表面积.

```
s[√(1/10 (5 + √5))]//Simplify
```

$(1+\sqrt{5})\pi$

```
Sign[s''[√(1/10 (5 + √5))]]
```

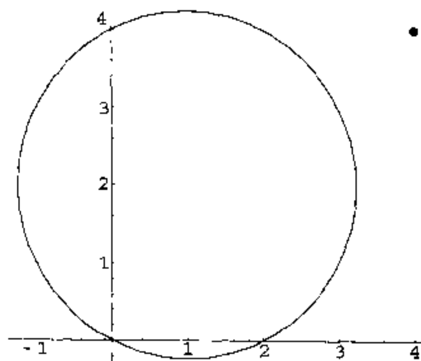
← 验证这是绝对最大值.

-1

8.18 求出圆 $x^2+y^2-2x-4y=0$ 上距离 $P(4,4)$ 最近和最远的点.

解

首先画一幅示意图.




```
<<Graphics`ImplicitPlot`
circ = ImplicitPlot[x^2 + y^2 - 2x - 4y == 0, {x, -5, 5},
```

```
            DisplayFunction →Identity];
p = Graphics[{PointSize[.02], Point[{4, 4}]}];
Show[circ, p, DisplayFunction → $DisplayFunction];
```

令 (x, y) 表示圆上一点.首先我们需要用 x 表示出 y.

```
Solve[x^2 + y^2 - 2x - 4y == 0, y]//Simplify
```

$\{\{y \to 2 - \sqrt{4 + 2x - x^2}\}, \{y \to 2 + \sqrt{4 + 2x - x^2}\}\}$

用 d2 表示从 (x, y)点到 (4, 4) 点距离的平方,下面计算它的最小值.从图上显然可知离 P 点最近的点在上半圆周.

```
y = 2 + Sqrt[4 + 2x - x^2];
d2[x_] = (x - 4)^2 + (y - 4)^2;
Solve[d2'[x] == 0]
```

$\{\{x \to \frac{1}{13}(13 + 3\sqrt{65})\}\}$

```
{x, y}/. x → 1/13(13 + 3 Sqrt[65])//Simplify
```

$\{1 + 3\sqrt{\frac{5}{13}}, 2 + 2\sqrt{\frac{5}{13}}\}$

```
%//N
```

{2.86052, 3.24035}

距离 P 点最远的点在下半圆周.

```
y = 2 - Sqrt[4 + 2x - x^2];
d2[x_] = (x - 4)^2 + (y - 4)^2;
Solve[d2'[x] == 0]
```

$\{\{x \to \frac{1}{13}(13 - 3\sqrt{65})\}\}$

```
{x, y}/. x → 1/13(13 - 3 Sqrt[65])//Simplify
```

$\{1 - 3\sqrt{\frac{5}{13}}, 2 - 2\sqrt{\frac{5}{13}}\}$

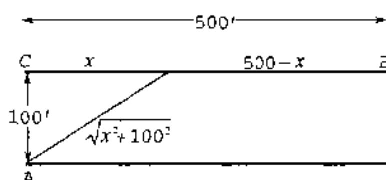
```
%//N
```

{-0.860521, 0.759653}

8.19 某地有一个电话公司计划在宽 100 英尺的河两边的 A 与 B 点之间架一条电话线,C 点为 A 点在河另一边的相对点,B 点到 C 点的距离为 500 英尺.然而,在水下架线的成本是在陆地上架线成本的 3 倍.在此条件下,公司应如何确定架线方案,从而使费用最小?

解 ☞

设在陆地上每架一英尺线的费用为 a,那么在水下架线一英尺的费用为 $3a$.这样总费用为 $c(x) = 3a\sqrt{x^2 + 100^2} + a(500 - x)$.当然,这时必定有 $0 \leqslant x \leqslant 50$.




```
c[x_] = 3a Sqrt[x^2 + 100^2] + a(500 - x);
```

```
Solve[c'[x] == 0, x]
{{x→25 √2}}
```

现在必须把对应于这个值的费用与在区间端点上的费用进行比较.

为了得到最小的费用,首先在水下把线架到距 C 点 $25\sqrt{2}$ 英尺的地方,然后从该点在陆地上架到 B 点.

```
c[0]//N
800.a
c[25 √2]//N
782.843a
c[500]//N
1529.71a
```

8.4 幂级数

最好处理的函数是多项式函数,因为它是连续函数,而且可以很容易地计算它的微分与积分.基于此,如果在实际问题中遇到了一个困难的函数,我们可以想办法用多项式逼近它.

如果知道了函数在单个点 a 的函数值与它的各阶导数值,那么函数可以用幂级数准确表示,然而这通常是一个无穷级数,在实际应用中必须截断.技巧就是使截断后的函数在 a 的某个邻域内比较准确地逼近给定函数.

下面这个所谓的泰勒(Taylor)级数给出了解析①函数 $f(x)$ 的表示.如果 $a=0$,这个级数就是马克劳林(Maclaurin)级数.

$$f(x) = \sum_{k=1}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k,$$

其中 $f^{(k)}(a)$ 表示 f 在 a 点的 k 阶导数.如果 $k=0$,它就表示 $f(a)$.

如果截断这个无穷幂级数,忽略所有次数高于 n 的项,那么就得到了 f 在 a 点的 n 次泰勒多项式.记这个多项式为 $p_n(x)$.如果 $a=0$,这个多项式也称为马克劳林多项式.

例 17 为了得到函数 $f(x)=e^x$ 的 5 次马克劳林多项式,我们要利用 Sum 命令或者 **BasicInput** 模板上的 $\sum$ 符号.下面给出了三种不同的生成这个多项式的方法.

```
f[x_] = Exp[x];
(a) Sum[(D[f[x], {x, k}]/. x→0)/k! * x^k, {k, 0, 5}]
```

$1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}$

(b) $\sum_{k=0}^{5} \frac{\partial_{\{x,k\}} f[x] /. x \to 0}{k!} x^k$

$1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}$

(c) $\sum_{k=0}^{5} \frac{\text{Derivative}[k][f][0]}{k!} x^k$

$1+x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\frac{x^5}{120}$

在 Mathematica 中有一条方便的命令,用来构造逼近一个函数的级数.

■ **Series[f[x], {x, a, n}]** 生成一个 SeriesData 对象,表示 f[x]在 a 点的次数为 n 的泰勒多项式.

---

①实变量的解析函数是指它有泰勒级数展开.在实际应用中遇到的大多数函数都是这种类型的;然而,即使函数具有各阶导数,它也不一定是解析的.

例 18

```
f[x_] = Exp[x];
Series[f[x], {x, 0, 5}]
```

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + o[x]^6$$

在上面展开式中的符号 $o[x]^6$ 表示在(无穷)展开中被忽略项的阶. $o[x]^6$ 意味着被忽略项关于 x 的次数 $\geqslant 6$.

SeriesData 对象表示幂级数,它没有数值. 如果要计算由 Series 命令给出的级数的值,就会遇到麻烦.

例 19

```
f[x_] = Exp[x];
p[x_] = Series[f[x], {x, 0, 5}]
```

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + o[x]^6$$

```
p[1]
SeriesData::ssdn : Attempt to evaluate a series at the number 1; returning Indeterminate.
```

(试图计算级数在数 1 的值,返回不定值.)

```
Indeterminate
```

为了查看 SeriesData 对象的形式,可以利用 **InputForm** 命令.

■ **InputForm[表达式]** 以适合于作为 Mathematica 输入的形式显示表达式.

例 20

```
f[x_] = Exp[x];
s = Series[f[x], {x, 0, 5}];
InputForm[s]
SeriesData[x, 0, {1, 1, 1/2, 1/6, 1/24, 1/120}, 0, 6, 1]
```

为了把幂级数转化为可以计算其值的形式,可以用函数 **Normal** 把它转化为普通的多项式.

■ **Normal[级数]** 返回可以计算其值的级数的多项式表示. 这时 $o[x]^n$ 被忽略.

例 21

```
f[x_] = Exp[x];
s = Series[f[x], {x, 0, 5}]
```

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + o[x]^6$$

```
p[x_] = Normal[s]
```

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120}$$

```
p[1]
```

$$\frac{163}{60}$$

> Normal 已把 SeriesData 对象转化为普通的多项式,后者的值是可计算的.

在前面例子中得到的数为 163/60,它近似等于 2.71667. 把这个数与(已知)值 $e \approx 2.71828$ 比较,可知近似值的误差很小. 当多项式的次数增大时,这个误差就会趋于消失. 下面这个例子说明了这种情形.

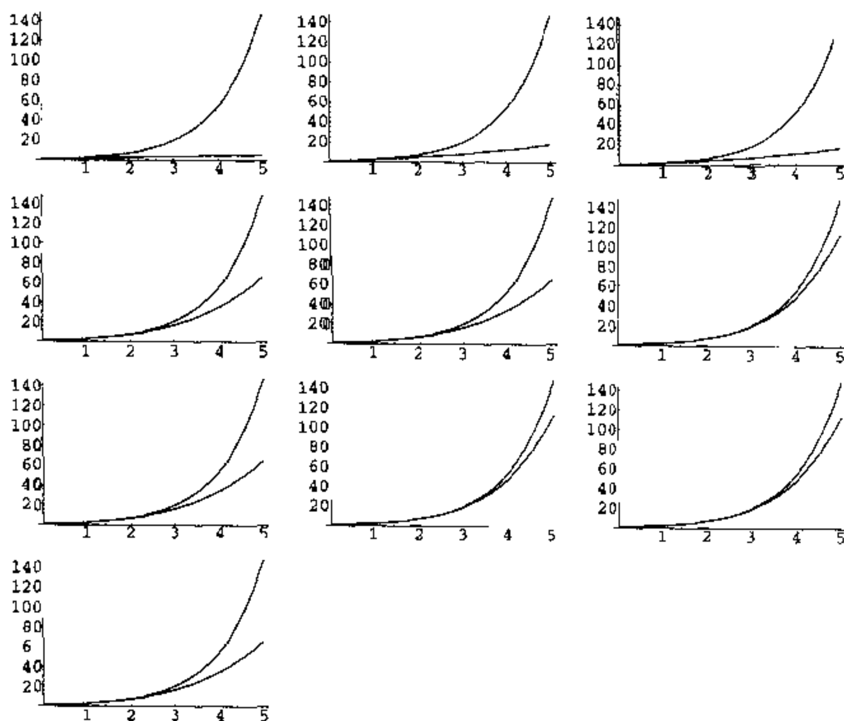
例 22

```
f[x_] = Exp[x];
exactvalue = f[1];
p[n_] := Normal[Series[f[x], {x, 0, n}]]/.x→1
data = Table[{n, N[p[n]], N[Abs[p[n] - exactvalue]]}, {n, 1, 10}];
TableForm[data, TableHeadings → {None, {"n", "p(1)", "Error"}}]
```

n	p(1)	Error
1	2.	0.718282
2	2.5	0.218282
3	2.66667	0.0516152
4	2.70833	0.0099485
5	2.71667	0.00161516
6	2.71806	0.000226273
7	2.71825	0.0000278602
8	2.71828	3.05862×10^{-6}
9	2.71828	3.02886×10^{-7}
10	2.71828	2.73127×10^{-8}

例 23 为了更进一步地具体查看幂级数收敛的情形,下面构造一个动画演示收敛于 e^x 的马克劳林多项式序列.考虑区间为 [0, 5].一旦生成了图像,选择图形单元括号,并进入 **Cell ⇒Animate Selected Graphics**,就可以启动动画演示.下面列出的是各帧图像.

```
f[x_] = Exp[x];
p[n_, x_] := Normal[Series[f[t], {t, 0, n}]]/. t→x
Do[Plot[{p[n, x]}, {x, 0, 5}, PlotRange→{0, Exp[5]}], {n,1,10}]
```



如果只需要级数中某一项的系数,那么可以用命令 **SeriesCoefficient**.这时实际的级数可能很长,但并不需要完整显示出来.SeriesCoefficient 类似于多项式时的 Coefficient 命令.

■ **SeriesCoefficient[级数, n]** 返回幂级数中次数为 n 的项的系数.

例 24
```
f[x_] = Exp[x];
s = Series[f[x], {x, 0, 10}];
SeriesCoefficient[s, 10]
```
$\frac{1}{3628800}$

习题解答

8.20 利用直接求和的方法给出函数 $f(x)=\tan^{-1}x$ 的 10 次马克劳林多项式，然后再用 Series 命令达到同样的目的.

解 ☞

```
f[x_] = ArcTan[x];
```
$\sum_{k=0}^{10}\frac{\text{Derivative[k][f][0]}}{k!}x^k$

$x-\frac{x^3}{3}+\frac{x^5}{5}-\frac{x^7}{7}+\frac{x^9}{9}$

```
Series[f[x], {x, 0, 10}]
```
$x-\frac{x^3}{3}+\frac{x^5}{5}-\frac{x^7}{7}+\frac{x^9}{9}+o[x]^{11}$

8.21 用 $x-2$ 的幂次组合给出 x^5 的表示.

解 ☞

```
Series[x^5, {x, 2, 5}]//Normal
```
$32+80(-2+x)+80(-2+x)^2+40(-2+x)^3+10(-2+x)^4+(-2+x)^5$

8.22 构造函数 $\sqrt{x}$ 在 $a=1$ 点的 5 次泰勒多项式，并用它给出 $\sqrt{3/2}$ 的近似值.

解 ☞

```
p[x_] = Series[√x, {x, 1, 5}]//Normal
```
$1+\frac{1}{2}(-1+x)-\frac{1}{8}(-1+x)^2+\frac{1}{16}(-1+x)^3-\frac{5}{128}(-1+x)^4+\frac{7}{256}(-1+x)^5$

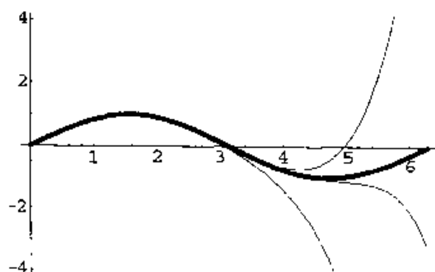
```
approx = p[3/2]//N
1.22498
exact = √3/2//N
1.22474
Abs[approx - exact]
0.000230715          ←这给出近似的绝对误差.
```

8.23 令 $f(x)=\sin x$，构造它的次数分别为 7, 9, 11 的马克劳林多项式，然后在同一坐标系中画出函数 f 以及三个多项式在 $0\leqslant x\leqslant 2\pi$ 上的图形，以观察它们的行为.

解 ☞

```
f[x_] = Sin[x];
p7[x_] = Series[f[x], {x, 0, 7}]//Normal;
p9[x_] = Series[f[x], {x, 0, 9}]//Normal;
p11[x_] = Series[f[x], {x, 0, 11}]//Normal;
```

```
Plot[{f[x], p7[x], p9[x], p11[x]}, {x, 0, 2π},
    PlotStyle →{Thickness[.01], Thickness[.001],
                Thickness[.001], Thickness[.001]}];
```



当多项式的次数增加，多项式就更好地逼近函数 $f(x)=\sin x$.

8.24 令 $f(x)=\sin x$，构造它的 11 次马克劳林多项式，然后再构造一个误差函数，并计算它从 $x=0$ 到 $x=6$ 按单位步长变化时的值. 把结果用表格的形式列出来，并说明当 x 远离 0 时误差的变化趋势.

解 ☞

```
f[x_] = Sin[x];
p11[x_] = Normal[Series[f[x], {x, 0, 7}]];
error[x_] = Abs[f[x] - p11[x]];
errorvalues = Table[{x, error[x]}, {x, 0, 6, 1.}];
TableForm[errorvalues, TableHeadings →{None, {"x", " error[x]"}}]
```

x	error[x]
0	0
1.	1.59828×10^{-10}
2.	1.29086×10^{-6}
3.	0.000245414
4.	0.0100021
5.	0.174693
6.	1.78084

当 x 远离 0 点，误差变得越来越大.

8.25 令 $f(x)=\sin x$. 构造次数分别为 1, 3 ,5, 7, 9 的马克劳林多项式，并计算它们在 $x=1$ 时的值. 确定近似误差，并用表格列出结果.

解 ☞

```
f[x_] = Sin[x];
exactvalue = f[1];
p[n_] := Normal[Series[f[x], {x, 0, n}]]/. x →1
data = Table[{n, N[p[n]], N[Abs[p[n] - exactvalue]]}, {n, 1, 9, 2}];
TableForm[data, TableHeadings →{None, {"n", " p(1)", "Error"}}]
```

n	p(1)	Error
1	1.	0.158529
3	0.833333	0.00813765

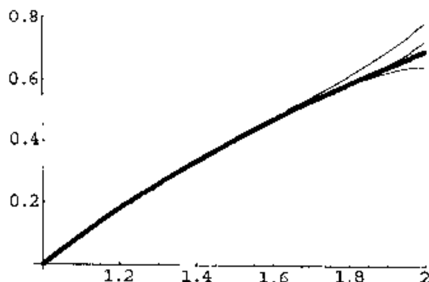
当 n 变大时，误差变得越小.

5	0.841667	0.000195682
7	0.841468	2.73084×10^{-6}
9	0.841471	2.48923×10^{-8}

8.26 令 $f(x)=\ln x$，构造它在 $a=1$ 处的 5 次，10 次与 15 次泰勒多项式．然后在同一坐标系中画出函数 $f(x)$ 与三个多项式在 $1\leqslant x\leqslant 2$ 上的图形．

解 ☞

```
f[x_] = Log[x];
p5[x_] = Series[f[x], {x, 1, 5}]//Normal;
p10[x_] = Series[f[x], {x, 1, 10}]//Normal;
p15[x_] = Series[f[x], {x, 1, 15}]//Normal;
Plot[{f[x], p5[x], p10[x], p15[x]}, {x, 1, 2},
    PlotStyle →{Thickness[.01], Thickness[.001],
               Thickness[.001], Thickness[.001]}];
```



8.27 令 $f(x)=\ln x$，构造它在 $a=1$ 时的 5 次泰勒多项式．然后构造一个误差函数，并计算它在 $x=1$ 变到 $x=2$，步长为 0.1 时的值．把结果以表格的形式显示出来，并说明当 x 远离 1 时，误差的变化趋势．

解 ☞

```
f[x_] = Log[x];
p5[x_] = Normal[Series[f[x], {x, 1, 5}]];
error[x_] = Abs[f[x] - p5[x]];
errorvalues = Table[{x, error[x]}, {x, 1, 2, .1}];
TableForm[errorvalues, TableHeadings →{None, {"x", " error[x]"}}]
```

x	error[x]
1	0
1.1	1.53529×10^{-7}
1.2	9.10987×10^{-6}
1.3	0.0000967355
1.4	0.000509097
1.5	0.00182656
1.6	0.00514837
1.7	0.0122941
1.8	0.026016
1.9	0.0502191
2.	0.0901862

当 x 远离 1 时，误差变得越来越大．

8.28　令 $f(x)=\ln x$. 构造次数分别为 1, 2, 3, …, 10 的马克劳林多项式, 并计算它们在 $x=1.5$处的值. 确定近似值的误差, 并用表格列出它们.

解 ☞

```
f[x_] = Log[x];
exactvalue = f[1.5];
p[n_] := Normal[Series[f[x], {x, 1, n}]]/. x→1.5
data = Table[{n, N[p[n]], N[Abs[p[n] - exactvalue]]}, {n, 1, 10}];
TableForm[data, TableHeadings→{None, {"n","p(1.5)", "Error"}}]
```

n	p(1.5)	Error
1	0.5	0.0945349
2	0.375	0.0304651
3	0.416667	0.0112016
4	0.401042	0.00442344
5	0.407292	0.00182656
6	0.404688	0.000777608
7	0.405804	0.000338463
8	0.405315	0.000149818
9	0.405532	0.000067196
10	0.405435	0.0000304603

8.29　在 $\sin(x^2+1)$ 的马克劳林级数中第 20 项的系数是多少?

解 ☞

```
s = Series[Sin[x^2 + 1], {x, 0, 20}];
SeriesCoefficient[s, 20]
```

$-\frac{\mathrm{Sin}[1]}{3628800}$

第九章 积分的计算

9.1 反导数

函数 f 的反导数就是满足 $F'(x)=f(x)$ 的函数 F. 命令 **Integrate** 可以得到反导数. 然而要注意在答案中并不包含积分常数 C.

■ **Integrate[f[x], x]** 计算反导数(不定积分) $\int f(x)dx$. 也可以用 **BasicInput** 模板上的 $\int \square \, d \square$ 符号.

Mathematica 可以计算在标准积分表中能找到的初等函数的反导数,但是如果不能用初等函数表示反导数,这个软件就会尝试用特殊函数表示反导数.如果这还不行的话,它就会返回无法计算的积分.

例 1 $\int$ **x²Exp[x]Sin[x] d x** 或者 **Integrate[x ^2 Exp[x] Sin[x], x]**

$-\frac{1}{2}e^{x}(-1+x^{2})\text{Cos}[x]+\frac{1}{2}e^{x}(-1+x^{2})\text{Sin}[x]$

例 2 $\int$ **Sin[x²] d x** 或者 **Integrate[Sin[x ^2], x]**

$\sqrt{\frac{\pi}{2}}\text{FresnelS}\left[\sqrt{\frac{2}{\pi}}x\right]$

这个积分没有简单的反导数,因此 Mathematica 用 Fresnel 正弦积分表示它:

$$S[x]=\int_0^x \sin\left(\pi\frac{t^2}{2}\right)dt$$

例 3 $\int$ **Sin[Sin[x]] d x** 或者 **Integrate[Sin[Sin[x]], x]**

$\int \text{Sin}[\text{Sin}[x]]\,dx$ ← Mathematica 不能计算这个反导数.

当要处理包含参数的一般反导数时,必须仔细一些.

例 4 $\int$ **xⁿ d x**

$\frac{x^{1+n}}{1+n}$

当然,这个结果只有在 $n \neq -1$ 时成立,但是如果指定了 n 的具体值,Mathematica 就知道如何计算了.

n = -1;

$\int$ **xⁿ d x**

Log[x]

习题解答

9.1 计算不定积分 $\int \sqrt{x}\,\mathrm{d}x$.

解 ☞

$\int \sqrt{\mathbf{x}}\,\mathbf{d\,x}$ 或者 `Integrate[√x, x]`

$\frac{2x^{3/2}}{3}$

9.2 计算不定积分 $\int \sqrt{a^2+x^2}\,\mathrm{d}x$.

解 ☞

$\int \sqrt{\mathbf{a}^2+\mathbf{x}^2}\,\mathbf{d\,x}$ 或者 `Integrate[√(a²+x²), x]`

$\frac{1}{2}x\sqrt{a^2+x^2}+\frac{1}{2}a^2\mathrm{Log}\left[x+\sqrt{a^2+x^2}\right]$

9.3 计算不定积分 $\int \frac{1}{\sqrt{u^2-a^2}}\mathrm{d}u$.

解 ☞

$\int \frac{1}{\sqrt{\mathbf{u}^2-\mathbf{a}^2}}\,\mathbf{d\,u}$ 或者 `Integrate[1/Sqrt[u² - a²], u]`

$\mathrm{Log}\left[u+\sqrt{-a^2+u^2}\right]$

9.4 计算不定积分 $\int \tanh x\,\mathrm{d}x$.

解 ☞

$\int$ `Tanh[x]` d x 或者 `Integrate[Tanh[x], x]`

Log[Cosh[x]]

9.5 计算(a) $\int f'(x)\mathrm{d}x$；(b) $\int g'(f(x))f'(x)\mathrm{d}x$.

(a) $\int$ `f'[x]` d x

f[x]

(b) $\int$ `g'[f[x]]f'[x]` d x

g[f[x]]

9.6 构造 $\int \sin^n x\,\mathrm{d}x$ 在 $n=1, 2, \cdots, 10$ 时的积分表.

解 ☞

```
anti[n_]: = ∫ Sin[x]^n d x
Table[{n, anti[n]//Together}, {n, 1, 10}]//TableForm
```

1　−Cos[x]

2　$\frac{1}{4}(2x-\mathrm{Sin}[2x])$

3 $\frac{1}{12}(-9\text{Cos}[x]+\text{Cos}[3x])$

4 $\frac{1}{32}(12x-8\text{Sin}[2x]+\text{Sin}[4x])$

5 $\frac{1}{240}(-150\text{Cos}[x]+25\text{Cos}[3x]-3\text{Cos}[5x])$

6 $\frac{1}{192}(60x-45\text{Sin}[2x]+9\text{Sin}[4x]-\text{Sin}[6x])$

7 $\frac{1}{2240}(-1225\text{Cos}[x]+245\text{Cos}[3x]-49\text{Cos}[5x]+5\text{Cos}[7x])$

8 $\frac{1}{3072}(840x-672\text{Sin}[2x]+168\text{Sin}[4x]-32\text{Sin}[6x]+3\text{Sin}[8x])$

9 $\frac{1}{80640}(-39690\text{Cos}[x]+8820\text{Cos}[3x]-2268\text{Cos}[5x]+405\text{Cos}[7x]-35\text{Cos}[9x])$

10 $\frac{1}{10240}(2520x-2100\text{Sin}[2x]+600\text{Sin}[4x]-150\text{Sin}[6x]+25\text{Sin}[8x]-2\text{Sin}[10x])$

9.2 定积分

定积分可以用两种方法计算,一种是准确计算,即用微积分基本定理进行计算,另一种是近似计算,即用数值方法计算积分.可以指导 Mathematica 选择具体的计算方法,方法就是选用下述两条命令:

- **Integrate[f[x], {x, a, b}]** 只要有可能,就计算出 $\int_a^b f(x)\mathrm{d}x$ 的准确值.可以用 **BasicInput** 模板上的 $\int_{\square}^{\square}\square\,\mathrm{d}\square$ 符号实现同样的功能.
- **NIntegrate[f[x], {x, a, b}]** 严格利用数值方法计算 $\int_a^b f(x)\mathrm{d}x$ 的近似值.

NIntegrate 利用自适应算法计算积分的近似值,它对积分区间进行分割,直到达到指定的准确度为止.实际上这里对区间进行递归分割,直到达到 **AccuracyGoal** 或者 **PrecisionGoal** 要求为止.

- **AccuracyGoal** 选项指定在最终结果中小数点右边应有多少位小数.默认值为 **AccuracyGoal → Infinity**,这规定不要用准确度作为终止数值算法的依据.
- **WorkingPrecision** 选项指定在中间计算过程中要保留多少位精确位数.默认值为 **WorkingPrecision → $ WorkingPrecision**,它通常设为 16.
- **PrecisionGoal** 指定在最终结果中应有多少位准确数字.默认设置为 **PrecisionGoal → Automatic**, **PrecisionGoal** 值为 **WorkingPrecision** 减去 10.

另外还有许多选项,可以控制实现的算法达到指定的精确度,但这里不予讨论.这些选项对讨论"病态"函数的积分(如 $\int_{.0001}^{1}\sin\left(\frac{1}{x}\right)\mathrm{d}x$ 或者 $\int_{-1000}^{1000}e^{-x^2}\mathrm{d}x$)非常有用,其使用细节,有兴趣的读者可以查阅 Mathematica 的帮助文件.

命令序列 **N[Integrate[f[x], {x, a, b}]]** 或者 $\int_a^b$**f[x] d x//N** 在有可能的情况下,先计算反导数,然后应用微积分基本定理计算积分的值.如果这个方法不行的话,它就会自动调用 **NIntegrate[f[x], {x, a, b}]**.

例5 为了计算 $\int_0^1 x e^x \sin x \mathrm{d}x$，输入命令

```
Integrate[x Exp[x] Sin[x], {x, 0, 1}]
```

$-\frac{1}{2}+\frac{1}{2}\mathrm{e}\,\mathrm{Sin}[1]$

我们也可以借助于 **BasicInput** 模板,采用另外一种表示.

$\int_0^1 \mathbf{x\ Exp[x]Sin[x]\ d\ x}$

如果希望给出数值近似值,可以输入

$\int_0^1 \mathbf{x\ Exp[x]Sin[x]\ d\ x//N}$ 或者 `N[Integrate[x Exp[x] Sin[x], {x, 0, 1}]]`

0.643678

这里,首先计算 $x e^x \sin x$ 的反导数,然后计算它在0与1上的函数值差.如果更希望采有一种数值积分方法,那么可以输入:

```
NIntegrate[x Exp[x] Sin[x], {x, 0, 1}]
```

0.643678

例6 假设我们要得到定积分 $\int_0^1 \sin(\sin x)\mathrm{d}x$ 的有20位有效数字的近似值,可以输入如下指令.

$\mathbf{N[\int_0^1 Sin[Sin[x]]\ d\ x,20]}$

0.43060610312069060491

这时 Mathematica 就会自动调整 WorkingPrecision 与 PrecisionGoal 的值,以达到所要求的精度.

Mathematica 也可以处理某些广义积分.第I类型的广义积分是指积分的一个或两个积分限为无穷.定义 $\int_a^{\infty} f(x)\mathrm{d}x = \lim_{t\to\infty}\int_a^t f(x)\mathrm{d}x$, $\int_{-\infty}^{b} f(x)\mathrm{d}x = \lim_{t\to\infty}\int_{-t}^{b} f(x)\mathrm{d}x$,前提条件是这两个极限存在.这样的积分称为收敛积分.如果 $\int_{-\infty}^{a} f(x)\mathrm{d}x$ 与 $\int_a^{\infty} f(x)\mathrm{d}x$ 都存在,我们就定义 $\int_{-\infty}^{\infty} f(x)\mathrm{d}x = \int_{-\infty}^{a} f(x)\mathrm{d}x + \int_a^{\infty} f(x)\mathrm{d}x$.

例7 $\int_0^{\infty} \mathbf{E^{-x}\ d\ x}$

1

例8 $\int_0^{\infty} \mathbf{x\ d\ x}$ ← 这个积分是发散的.

∞

例9 $\int_{-\infty}^{\infty} \frac{1}{1+x^2}\,\mathbf{d\ x}$

π

第I型广义积分的值可以与在积分式中的参数有关.选项 **Assumptions** 可以给这些参数加上取值条件.

- **Assumptions →条件**指定在积分式中参数取值要满足的条件.

例 10　对于广义积分 $\int_1^{\infty} x^n \mathrm{d}x$, 如果 $n < -1$, 则积分收敛, 否则积分发散.

```
Integrate[x^n, {x, 1, ∞}, Assumptions →n < -1]
```

$\frac{1}{-1-n}$

第 II 型广义积分是指函数在积分区间上不连续. 如果 f 在 $[a, b)$ 上连续, 但在 b 点不连续, 就定义 $\int_a^b f(x)\mathrm{d}x = \lim\limits_{t\to b^-}\int_a^t f(x)\mathrm{d}x$; 而如果 f 在 $(a, b]$ 上连续, 在 a 点不连续, 就定义 $\int_a^b f(x)\mathrm{d}x = \lim\limits_{t\to a^+}\int_t^b f(x)\mathrm{d}x$. 如果这两个极限存在的话, 我们称广义积分是收敛的. 如果 f 在 $c \in [a, b]$ 上不连续, 而 $\int_a^c f(x)\mathrm{d}x$ 与 $\int_c^b f(x)\mathrm{d}x$ 都收敛, 则定义 $\int_a^b f(x)\mathrm{d}x = \int_a^c f(x)\mathrm{d}x + \int_c^b f(x)\mathrm{d}x$.

例 11　$\int_0^1$ **Log[x] d x**

-1

例 12　$\int_{-2}^3 \frac{1}{x}$ **d x** 或者 **Integrate[1/x, {x, -2, 3}]**

```
Integrate::idiv:
Integral of 1/x does not converge on {-2, 3}.
```

($\frac{1}{x}$ 在 [-2, 3] 上的积分不收敛.)

$\int_{-2}^3 \frac{1}{x}$ d x

有时候, 即使积分发散, 但它的柯西主值是存在的.

- **PrincipalValue →True** 指定计算的是积分的柯西主值.

下面这个例子讲解了柯西主值的含义.

例 13　考虑例 12 中的积分 $\int_{-2}^3 \frac{\mathrm{d}x}{x}$. 由于被积函数在 0 点不连续, 所以这是一个广义积分. 如果把积分分成两个积分的和: $\int_{-2}^0 \frac{\mathrm{d}x}{x} + \int_0^3 \frac{\mathrm{d}x}{x}$, 那么分别计算这两个积分可知它们都是发散的. 然而

$$\begin{aligned}\lim_{t\to0^+}\left[\int_{-2}^{-t}\frac{\mathrm{d}x}{x}+\int_t^3\frac{\mathrm{d}x}{x}\right] &= \lim_{t\to0^+}\left[\ln|x|\,\Big|_{-2}^{-t}+\ln|x|\,\Big|_t^3\right]\\ &= \lim_{t\to0^+}\left[\ln t-\ln2+\ln3-\ln t\right]\\ &= \ln3-\ln2\\ &= \ln\frac{3}{2}\end{aligned}$$

计算柯西主值的基本想法就是不分开计算两个积分，而是同时计算它们.

```
Integrate[1/x, {x, -2, 3}, PrincipalValue →True]
```

$\mathrm{Log}\left[\frac{3}{2}\right]$

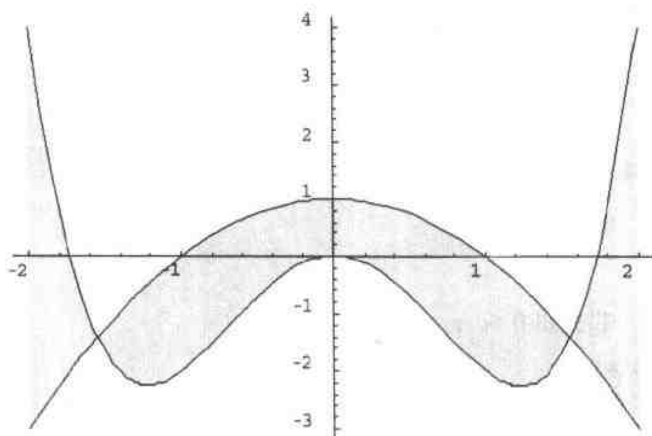
习题解答

9.7 计算由曲线 $f(x)=1-x^2$ 与 $g(x)=x^4-3x^2$ 围成的区域的面积.

解 ☞

```
f[x_] = 1 - x²;
g[x_] = x⁴ - 3x²;
<<Graphics`FilledPlot`
FilledPlot[{f[x], g[x]}, {x, -2, 2}];
intersectionpoints = Solve[f[x] == g[x]]
```

$\{\{x\to -i\sqrt{-1+\sqrt{2}}\},\{x\to i\sqrt{-1+\sqrt{2}}\},$
$\{x\to -\sqrt{1+\sqrt{2}}\},\{x\to \sqrt{1+\sqrt{2}}\}\}$



曲线的交点对应于这个方程的实根.

```
a = intersectionpoints[[3,1,2]];
b = intersectionpoints[[4,1,2]];
```

$\int_a^b (f[x]-g[x])\,dx$ //Simplify

$\frac{8}{15}\sqrt{1+\sqrt{2}}(4+\sqrt{2})$

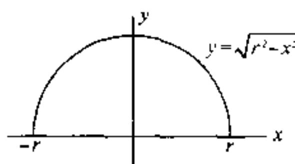
```
%//N
4.48665
```

9.8 由曲线 $y=f(x)$ 与区间$[a,b]$围成的区域绕 x 轴旋转得到的立体体积为 $\pi\int_a^b[f(x)]^2\mathrm{d}x$. 把半圆 $y=\sqrt{r^2-x^2}$, $-r\leqslant x\leqslant r$ 绕x 轴旋转，得到一个球，由前面的公式计算球的体积.

解 ☞

```
y = √(r² - x²);
π∫_{-r}^{r} y² d x
```

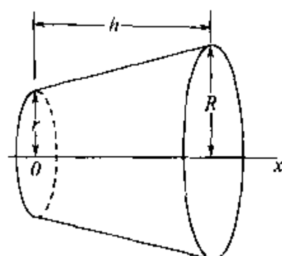
$\frac{4\pi r^3}{3}$



9.9 计算高为 h，上下底半径分别为 r 与 R 的圆台的体积，并由该公式给出底面半径为 R，高为 h 的圆锥的体积.

解 ☞

圆台是由连接 (0, r) 与 (h, R) 的直线段以及 x 轴围成的区域绕 x 轴旋转得到的立体.这条线段的方程为 $y=\frac{R-r}{h}x + r$，因此立体的体积为 $\pi\int_0^h y^2\mathrm{d}x$.




```
y = (R - r)/h x + r;
π∫_0^h y² d x//Factor
```

$\frac{1}{3}h\pi(r^2 + rR + R^2)$

```
%/.r → 0
```

◂ 令 $r=0$ 可得到圆锥.

$\frac{1}{3}h\pi R^2$

9.10 由 $f(x)$, $a \leqslant x \leqslant b$ 表示的一段弧线的长度为 $L=\int_a^b \sqrt{1+[f'(x)]^2}\mathrm{d}x$. 由此公式计算正弦曲线一拱的近似长度.

解 ☞

正弦曲线的一拱是由 $0 \leqslant x \leqslant \pi$ 确定的.

```
f[x_] = Sin[x];
NIntegrate[Sqrt[1 + f'[x]^2], {x, 0, Pi}]
3.8202
```

9.11 积分中值定理指的是如果 f 在闭区间[a, b]上连续，那么存在一个介于 a 与 b 之间的数 c，使得 $\int_a^b f(x)dx = f(c)(b-a)$. 对于区间[1, 2]上的函数 $f(x)=\ln x$，求出满足中值定理的值 c.

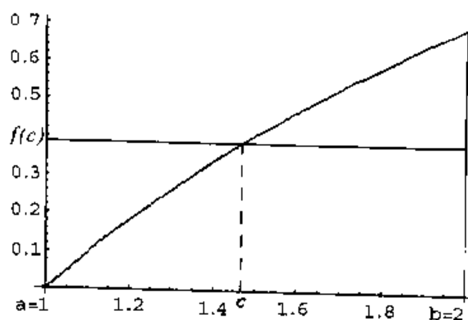
解 ☞

```
f[x_] = Log[x];
a = 1; b = 2;
Solve[∫_a^b f[x] d x == f[c](b - a), c]//Simplify
```

$\{\{c \to \frac{4}{e}\}\}$

```
%//N
{{c → 1.47152}}
```

> 在 x 轴上面，曲线下面的区域的面积，等于由 c 确定的矩形的面积.



9.12　把一个物体从 a 拖到 b，施加的力为变力 $f(x)$，那么所做的功为 $\int_a^b f(x)dx$．根据胡克定律，为了把一个弹簧拉长，所需的力正比于弹簧相应于自然长度的变化．如果一个弹簧的自然长度为 10 cm，把它拉长 5 cm 需要外力 50 牛顿，那么把弹簧从 10 cm 拉长到 15 cm 需要做多少功？

解

由胡克定律，$f(x)=kx$．由于为了拉长 5 cm(0.05 m)，需要外力 40 牛顿，所以有 $40=0.05k$．

```
k = 40/0.05;
f[x_] = k x;
work = ∫_0^0.05 f[x] d x
1.
```

做功 1 焦耳．

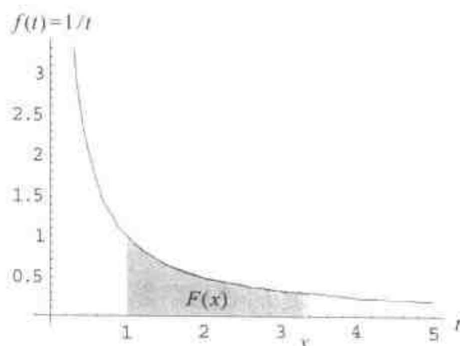
9.3　由积分定义的函数

如果 f 为区间 $[a,\ b]$ 上的连续函数，那么我们可以如下定义一个新的函数

$$F(x)=\int_a^x f(t)\mathrm{d}t.$$

直观地说，如果 $f(x)\geqslant 0$，那么当 $x\geqslant a$ 时，$F(x)$ 表示由 $f(t)$ 与 t 轴围成的介于 a 到 x 之间的区域的面积；如果 $x<a$，表示这个面积的相反数．由微积分(第二)基本定理可知：(i) F 在 $(a,\ b)$ 上可微，而且(ii) $F'(x)=f(x)$ 对所有的 $x\in(a,\ b)$ 成立．

例 14　令 $f(x)=1/x$, $x>0$．假设 $x\geqslant 1$，那么下图中的阴影部分的面积就是 $F(x)$．学过微积分的人都知道，这个积分把 $F(x)$ 定义为自然对数函数．Mathematica 也知道这一点．

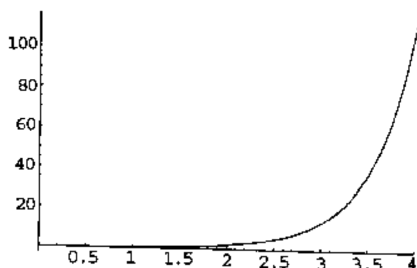



```
f[x_] = 1/x;
F[x_] = ∫_1^x f[t] d t
Log[x]
F[2]//N
0.693147
F[1/2]//N
-0.693147
```

例 15　函数 $f(x) = x^x$ 存在反导数，但这个反导数不能用初等函数表示出来. 然而 Mathematica 可以把这个函数认为是由积分定义的函数. 由于 $f(x)$ 的所有反导数之间差一个常数，我们把 $F(x)$ 定义成满足 $F(0) = 0$ 的反导数. 下面画出反导数在 $0 \leqslant x \leqslant 4$ 上面的图形.

```
f[x_] = x^x;
F[x_] = ∫_0^x f[t] dt
Plot[F[x], {x, 0, 4}]
```

← 通过把下限取为 0，从而使得 $F(0) = 0$.



习题解答

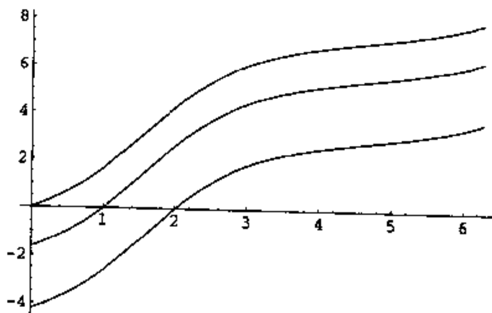
9.13　令 $F(x) = \int_1^x e^{\sin t} dt$，求出 $F'(x)$.

解 ☞

```
F[x_] = ∫_1^x Exp[Sin[t]] dt;
F'[x]
e^Sin[x]
```

这符合微积分第二基本定理.

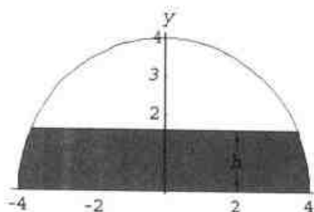
9.14　在同一坐标系中画出 $f(x) = e^{\sin x}$ 分别相应于 $F(0) = 0$, $F(1) = 0$ 与 $F(2) = 0$ 的三个反导数的图形. 绘图区间为 $0 \leqslant x \leqslant 2\pi$.



解 ☞

```
f[x_] = Exp[Sin[x]];
F1[x_] = ∫_0^x f[t] dt;
F2[x_] = ∫_1^x f[t] dt;
F3[x_] = ∫_2^x f[t] dt;
Plot[{F1[x], F2[x], F3[x]}, {x, 0, 2π}];
```

9.15　考虑右图所示的半圆 $x^2 + y^2 = 16$, $y > 0$，求出高度 h，使得阴影部分的面积等于整个半圆面积的一半.



解 ☞

从半圆的方程中用 y 解出函数 x.

```
Solve[x^2 + y^2 == 16, x]
```

```
{{x→ - √(16 - y²)}, {x→ √(16 - y²)}}
```

通过对 y 轴进行分割，并利用对称性，我们得到下述关于 $A(h)$ 的表示：$A(h) = 2\int_0^h x(y)dy$，其中 $x(y) = \sqrt{16 - y^2}$.

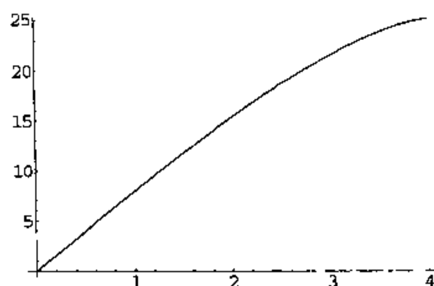
```
x[y_] = √(16 - y²);
A[h_] = 2 ∫₀ʰ x[y] d y;
```

下面先计算整个半圆的面积.(虽然我们已经知道圆的面积公式，但这不失为检验是否有错误的好方法.)

```
A[4]
8π
```

为了得到解的估计值，下面画出 $A(h)$ 的图形.

```
Plot[A[h], {h, 0, 4}];
```

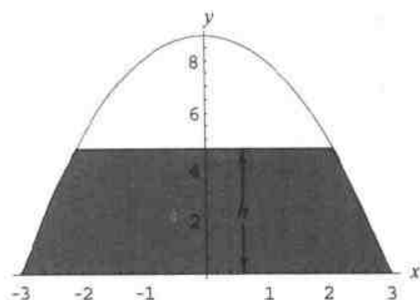


由图可见，半圆面积的一半 $4\pi\approx 12.5$ 对应于 h 的值靠近 1.5. 下面用 FindRoot 完成这道题目.

```
FindRoot[A[h] == 4π, {h, 1.5}]
{h → 1.61589}
```

9.16 下面给出的曲线为抛物线 $y = 9 - x^2$. 求出 h 的值使得阴影部分的面积是由曲线与 x 轴围成面积的 2/3.

解 ☞




```
Solve[y == 9 - x², x]
{{x→ - √(9 - y)}, {x→ √(9 - y)}}
x[y_] = √(9 - y);
A[h_] = 2 ∫₀ʰ x[y] d y;
totalarea = A[9]
36
Plot[A[h], {h, 0, 9}];
```



总面积的 2/3 为 24，大约对应于 h 等于 5 的地方.

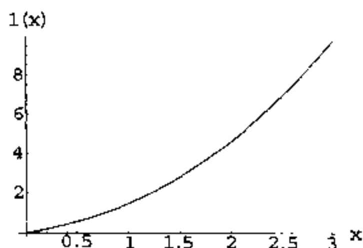
```
FindRoot[A[h] == (2/3)totalarea, {h, 5}]
{h → 4.67325}
```

9.17　在抛物线 $y = x^2$ 上哪一点离原点的曲线长度为 5?

解 ☞

函数 $f(x)$ 从 a 到 b 所定义曲线的长度为 $L = \int_a^b \sqrt{1 + [f'(x)]^2}dx$.

```
f[x_] = x^2;
l[x_] = ∫_0^x √(1 + f'[t]^2) dt;
Plot[l[x], {x, 0, 3}, AxesLabel→{"x", "l(x)"}];
```



由图上可知 L(2)≈5.

```
FindRoot[l[x] == 5, {x,2}]
{x → 2.08401}
{%[[1,2]], f[%[[1,2]]]}
{2.08401, 4.34308}
```

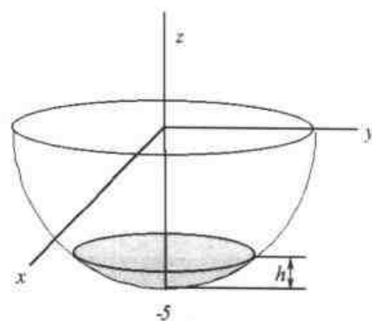
所求点为(2.08401,4.34308).

9.18　一个拌料用的碗是半径为 5 英寸的半球.计算其中盛 100 立方英寸液体时的高度.

解 ☞

在三维空间中半球的方程为 $x^2 + y^2 + z^2 = 25, z \leqslant 0$. 它与平面 $z = z_0$ 的交是圆 $x^2 + y^2 = 25 - z_0^2$, 其半径为 $\sqrt{25 - z_0^2}$, 面积为 $\pi(25 - z_0^2)$. 把这个表达式相应于 z_0 进行积分, 就得到阴影部分的体积为

$$V(h) = \pi\int_{-5}^{-5+h}(25 - z^2)\mathrm{d}z.$$




```
v[h_] = π ∫_{-5}^{-5+h} (25 - z^2) d z
(5h^2 - h^3/3) π
```

作为一种检验步骤, 可以计算 v(5)以得到半球的体积.

```
v[5]
250π/3
```

半径为 r 的半球体积为 $\frac{2}{3}\pi r^2$.

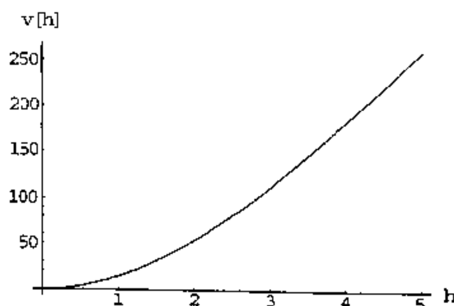
可以画出 h 的函数 v 的图形.

```
Plot[v[h], {h, 0, 5}, AxesLabel →{"h", "v[h]"}];
```

由于 v(h)是多项式函数, 所以可以用 NSolve 命令确定所求的近似值.(由上图可知, h 靠近 3.)

```
NSolve[v[h] == 100]
{{h → -2.34629}, {h → 2.79744}, {h → 14.5489}}
```

显然惟一的合理解为 $h = 2.79744$ 英寸.



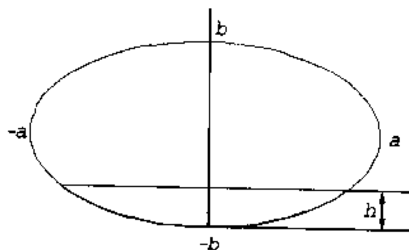
9.19 一个地下油罐是椭圆形截面的圆柱.这个油罐长 $l=20$ 英尺,椭圆的半长轴长 $a=10$ 英尺,半短轴长 $b=5$ 英尺.为了确定在油罐中还有多少油,用一个木棍插到油罐中,直到接触到油罐底为止,这时只要量一下棍上粘油部分有多长,就可以知道油的体积.如果在油罐中只有 500 立方英尺的油,木棍粘油部分应有多少?

解 ☞

椭圆的方程为$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$.首先把 x 定义成 y 的函数.

```
Solve[x^2/a^2 + y^2/b^2 == 1, x]
{{x -> -a Sqrt[1 - y^2/b^2]}, {x -> a Sqrt[1 - y^2/b^2]}}
```

现在就得到了油罐中截面的面积公式.这里利用对称性,只取正解,然后把面积加倍.




```
a = 10; b = 5;
x[y_] = a Sqrt[1 - y^2/b^2];
area[h_] = 2 ∫_{-b}^{-b+h} x[y] dy;
```

作为一种检验的方法,可以计算 area[0],area[b]与 area[2b]的面积.由椭圆$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$ 围成的区域面积为 πab.

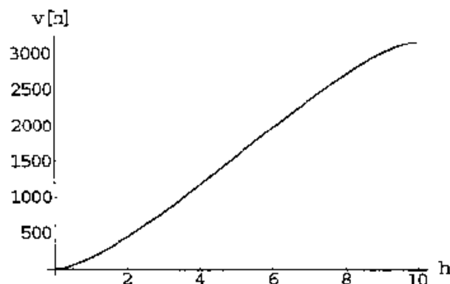
```
area[0]
0
area[b]
25π
area[2b]
50π
```

由于油罐的截面是均匀的,所以它的体积 = 长度 × 截面面积.

```
l = 20;
v[h_] = l area[h];
```

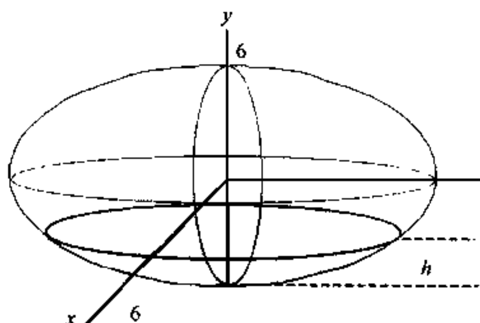
为了得到当油的体积为 500 立方英尺时木棍粘油的近似长度,下面先画出 v(h)的图形,给出 h 的一个估计值,然后再用 FindRoot 得到一个更精确的近似值.(我们不能同前一习题那样使用 NSolve 命令,因为这里 v(h)不是代数函数.)

```
Plot[v[h], {h, 0, 20}, AxesLabel -> {"h", "v[h]"}];
FindRoot[v[h] == 500, {h, 2}]
{h -> 2.1623}
```



从图上可知 V(2)≈500.

9.20 一个地下油罐是椭球体形状,其半轴长分别为 6 英尺,10 英尺与 6 英尺.(这个问题虽然相对于前一习题要困难一些,但更符合实际情形.)为了确定在油罐中还有多少油,用一个木棍插到油罐中,直到接触到油罐底为止,这时只要量一下棍上粘油部分有多长,就可以知道油的体积.如果在油罐中只有 500 立方英尺的油,木棍粘油部分应有多少?

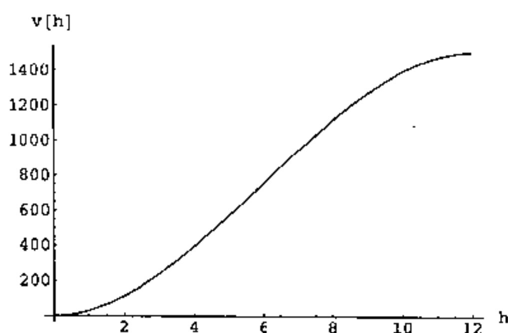


解答

椭球的方程为$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1$,其中 $a=6$, $b=10$, $c=6$.椭球与平面 $z=z_0$ 的交线为椭圆$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1-\frac{z_0^2}{c^2}$,它的面积为 z_0 的函数,可以计算出来.然后对这个面积相应于 z 积分就得到所求的体积.

为了计算椭圆的面积,我们要利用一个事实,那就是椭圆的面积等于 π 乘以它的半长轴与半短轴的积.把$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1-\frac{z_0^2}{c^2}$重写为等价的形式$\frac{x^2}{a^2\left(1-\frac{z_0^2}{c^2}\right)}+\frac{y^2}{b^2\left(1-\frac{z_0^2}{c^2}\right)}=1$,

可知其半长轴与半短轴为$a\sqrt{1-\frac{z_0^2}{c^2}}$与 $b\sqrt{1-\frac{z_0^2}{c^2}}$,因此椭圆的面积为 $\pi ab\left(1-\frac{z_0^2}{c^2}\right)$.代入这里 a, b 与 c 的具体值,得到 $60\pi\left(1-\frac{z_0^2}{6^2}\right)=\frac{5\pi}{3}(36-z_0^2)$.对它进行积分,就可以得到用 h 表示的油的体积公式: $V=\frac{5\pi}{3}\int_{-6}^{-6+h}(36-z^2)\mathrm{d}z$



v[h_] = $\frac{5\pi}{3}\int_{-6}^{-6+h}(36-z^2)\,d\,z$;

```
Plot[v[h], {h, 0, 12},
AxesLabel →{"h", "v[h]"}];
NSolve[v[h] == 500]
{{h → - 3.63858}, {h → 4.62871},
{h → 17.0099}}
```

因此木棍粘油的部分长度约为 4.62871 英尺.其它解都是增根.

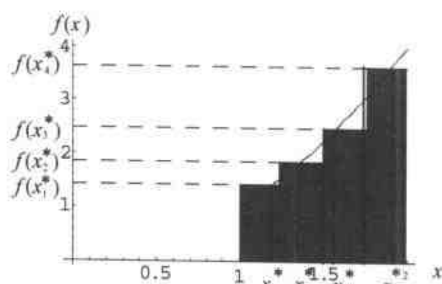
9.4 黎曼和

给定区间 I=[a, b],所谓对它的分割,就是一组子区间:

$$[x_0,\ x_1],[x_1,\ x_2],\cdots,[x_{n-1},\ x_n],$$

其中 $x_0=a, x_n=b$.如果取 x_i^* 为第 i 个子区间中的任一点,$\Delta x_i=x_i-x_{i-1}$表示第 i 个子区间的长度,$\|P\|=\max\limits_{1\leqslant i\leqslant n}\Delta x_i$,那么 f 在区间 I 上相应于 P 的黎曼和为 $\sum\limits_{i=1}^{n}f(x_i^*)\Delta x_i$.

如果 $f(x)\geqslant 0, a\leqslant x\leqslant b$,那么上述黎曼和就表示从 $x=a$ 到 $x=b$ 在 $f(x)$的图形下方,x 轴上方区域面积的近似值.下图给出的是函数 $f(x)=x^2$ 在区间 [1, 2] 上的一种黎曼和,这里给出的黎曼和是由等宽度的四个矩形的面积组成的.



在上图中由阴影部分围成的矩形面积就是黎曼和,它只是曲线下方区域面积的近似.然而,当取越来越多的矩形时,近似的效果就会越来越好,在曲线下方的面积就是这个过程的极限.

在大多数微积分教材中都把[a, b]上 $f(x)$ 的定积分定义为

$$\int_a^b f(x)dx = \lim_{\|P\|\to 0}\sum_{i=1}^{n}f(x_i^*)\Delta x_i.$$

条件 $\|P\|\to 0$ 保证当取越来越多的子区间时,所有子区间的长度都趋向于零.如果所有子区间的长度都相同,那么这个条件就等价于 $n\to\infty$.为了简单起见,我们下面只考虑等长度的子区间.然而从理论的角度来说,并不必一定是这样的.

例 16 下面考虑在区间[0, $\pi/2$] 上的函数 $f(x)=\sin x$.由于已知 $\int_0^{\pi/2}\sin x\,dx=1$, 所以很容易知道下述结果的逼近程度.

(a) 下面使用 100 个子区间,并取 x_i^* 为每个子区间的左端点.由于 $\sin x$ 在这个区间上是单调递增的,所以得到下方近似:

```
f[x_] = Sin[x];
a = 0; b = π/2; n = 100;
Δx = (b - a)/n;             ← 由于所有的 Δx 长度一样,因此不需加下标.
xstar[i_] = a + (i-1)Δx;
 n
 ∑  f[xstar[i]]Δx//N
i=1
0.992125
```

(b) 选择 x_i^* 为每个子区间的右端点,这样可以得到上方近似.

```
xstar[i_] = a + i Δx;
```

$$\sum_{i=1}^{n} \texttt{f[xstar[i]]}\Delta \texttt{x//N}$$

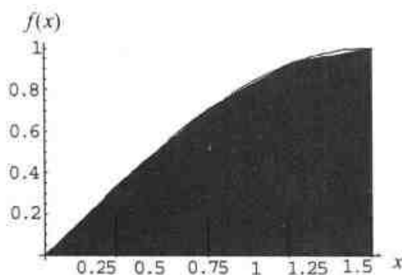
```
1.00783
```

(c) 为了改进上述近似效果,我们取 x_i^* 为每个子区间的中点.这样就得到了所谓的中点法则方法.

```
xstar[i_] = a + (i + .5) Δx;
```

$$\sum_{i=1}^{n} \texttt{f[xstar[i]]}\Delta \texttt{x//N}$$

```
1.00001
```

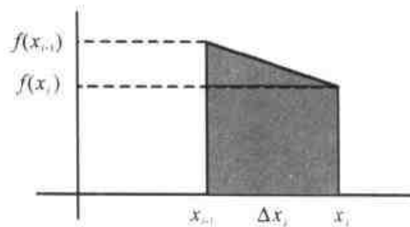


利用四个梯形的面积和近似 $\int_0^{\pi/2} \sin x \mathrm{d}x$.

显然,近似的效果得到提高.

另外一种简单的近似方法,就是梯形法则,它通过连接每个子区间两个端点对应于曲线上的点构成一个梯形,计算面积时使用的是这些梯形的面积和,而不是原来的矩形面积,从而提高近似的精度.

底边长为 Δx,两个侧边长分别为 A 与 B 的梯形面积为 $\frac{\Delta x}{2}(A+B)$,因此在第 i 个子区间 $[x_{i-1}, x_i]$ 中构造的梯形面积为 $\frac{\Delta x_i}{2}[f(x_{i-1})+f(x_i)]$.



把每个子区间上的梯形面积加起来,就得到整个曲边梯形的面积,即

$$\frac{\Delta x_1}{2}[f(x_0)+f(x_1)]+\frac{\Delta x_2}{2}[f(x_1)+f(x_2)]+\frac{\Delta x_3}{2}[f(x_2)+f(x_3)]+\cdots+\frac{\Delta x_n}{2}[f(x_{n-1})+f(x_n)].$$

如果所有子区间的长度相同,都为 Δx,那么上式简化为

$$\frac{\Delta x}{2}[[f(x_0)+f(x_1)]+[f(x_1)+f(x_2)]+[f(x_2)+f(x_3)]+\cdots+[f(x_{n-1})+f(x_n)]],$$

或者

$$\frac{\Delta x}{2}[f(x_0)+2f(x_1)+2f(x_2)+\cdots+2f(x_{n-1})+f(x_n)].$$

例 17

```
f[x_] = Sin[x];
a = 0; b = π/2; n = 100;
Δx = (b - a)/n;
x[i_] = a + i Δx;
```

$$\texttt{approximation} = \frac{\Delta \texttt{x}}{2}\left(\texttt{f[a]} + 2\sum_{i=1}^{n-1}\texttt{f[x[i]]} + \texttt{f[b]}\right) \texttt{//N}$$

```
0.999979
```

习题解答

9.21 分别使用下述方法计算 $f(x)=x\ln x$ 在区间 [1, 2] 上的黎曼和：

(a) 每个子区间的左端点；

(b) 每个子区间的右端点；

(c) 每个子区间的中点.

并把结果与 Mathematica 的近似值做一对比.

解 ☞

```
f[x_] = x Log[x];
a = 1; b = 2;
∫_a^b f[x] dx//N
0.636294                ←Mathematica 的近似结果.
n = 100;
Δx = (b - a)/n;
xstar[i_] = a + (i-1)Δx;
Sum[f[xstar[i]]Δx, {i, 1, n}]//N
0.629369                ←左端点.
xstar[i_] = a + i Δx;
Sum[f[xstar[i]]Δx, {i, 1, n}]//N
0.643232                ← 右端点.
xstar[i_] = a + (i + .5) Δx;
Sum[f[xstar[i]]Δx, {i, 1, n}]//N
0.636291                ←中点.
```

9.22 对于区间 [0, 1] 上的函数 $f(x)=x^2$，计算它对于 $n=2, 4, 8, 16, \cdots, 2^{20}$ 个子区间的黎曼上和与下和.并用积分 $\int_0^1 x^2 dx$ 解释近似的效果.

解 ☞

```
f[x_] = x^2;
a = 0; b = 1;
n = 2^m;
Δx = (b - a)/n;
nn = PaddedForm[n, 10];
temp1 = PaddedForm[N[Sum[f[a + (i - 1)Δx]Δx, {i, 1, n}]], {8,6}];
temp2 = PaddedForm[N[Sum[f[a + iΔx]Δx, {i, 1, n}]], {8,6}];
lst = Table[{nn, temp1, temp2}, {m, 1, 20}];
TableForm[lst, TableHeadings →
    {None, {"         n", "     lower", "     upper"}}]
```

n	lower	upper
2	0.125000	0.625000
4	0.218750	0.468750
8	0.273438	0.398438
16	0.302734	0.365234
32	0.317871	0.349121
64	0.325562	0.341187
128	0.329437	0.337250
256	0.331383	0.335289
512	0.332357	0.334311
1024	0.332845	0.333822
2048	0.333089	0.333578
4096	0.333211	0.333455
8192	0.333272	0.333394
16384	0.333303	0.333364
32768	0.333318	0.333349
65536	0.333326	0.333341
131072	0.333330	0.333337
262144	0.333331	0.333335
524288	0.333332	0.333334
1048576	0.333333	0.333334

当 n 变大时，下和增加，趋向于极限 1/3，同时上和减小，也趋向于极限 1/3，因为 $\int_0^1 x^2 dx = \frac{1}{3}$

9.23 利用梯形法则计算 $\int_0^1 e^{x^2} dx$ 的值，分别使用 10, 50, 100 个子区间，并与 Mathematica 给出的近似值比较.

解 ☞

```
f[x_] = Exp[x^2];
a = 0; b = 1;
∫_a^b f[x] dx//N
1.46265
Clear[n];
Δx = (b - a)/n;
x[i_] = a + i * Δx;
n = 10;
approximation = Δx/2 (f[a] + 2 Σ_{i=1}^{n-1} f[x[i]] + f[b])//N
1.46717
n = 50;
approximation = Δx/2 (f[a] + 2 Σ_{i=1}^{n-1} f[x[i]] + f[b])//N
1.46283
n = 100;
approximation = Δx/2 (f[a] + 2 Σ_{i=1}^{n-1} f[x[i]] + f[b])//N
1.4627
```

第十章 多元微积分运算

10.1 偏导数的计算

在第八章中讨论的 D,∂ 与 Derivative 命令实际上是用来计算偏导数的.当然,如果在函数中只有一个变量,那么偏导数就是通常的导数.但是如果有两个或多个变量,那么在计算导数时,除了指定的变量外,其它变量都认为是常数.

在下面的语法描述中,f 表示多元函数.

- **D[f, x]**或 ∂_x**f**(利用 **BasicInput** 模板输入)给出 ∂f/∂x,即 f 相应于变量 x 的偏导数.
- **D[f, {x, n}]** 或 $\partial_{\{x,n\}}$**f** 给出 $\partial^n f/\partial^n x$,即 f 相应于变量 x 的 n 阶偏导数.
- **D[f,** $x_1, x_2, \cdots, x_k$**]**或 $\partial_{x_1,x_2,\cdots,x_k}$**f** 给出混合偏导数 $\partial^k f/\partial x_1\partial x_2\cdots\partial x_k$.
- **D[f, {**x_1, n_1**}, {**x_2, n_2**}, …, {**x_k, n_k**}]**或 $\partial_{\{x_1,n_1\},\{x_2,n_2\},\cdots,\{x_k,n_k\}}$**f** 给出偏导数 $\dfrac{\partial^n f[x_1,x_2,\cdots,x_k]}{\partial x_1^{n_1}\partial x_2^{n_2}\cdots\partial x_k^{n_k}}$,其中 $n_1+n_2+\cdots+n_k=n$.

为了方便起见,在偏导数符号中可以插入不可见的逗号.插入方法就是输入三键序列[ESC] + , +[ESC].不可见逗号的作用与普通逗号类似,但不显示出来.

例 1 **D[x²y³z⁴, x]**

$2xy^3z^4$

∂_y**(x²y³z⁴)** ←这里的括号非常重要,为什么呢?

$3x^2y^2z^4$

D[x²y³z⁴, {z, 2}]

$12x^2y^3z^2$

$\partial_{x,y}$**(x²y³z⁴)**

$6xy^2z^4$

例 2 令 $f(x, y) = x^5y^7$.下面的指令可以计算 $f_{xxxyyyy}$.

```
f[x_, y_] = x^5 y^7;
D[f[x,y], {x, 3}, {y, 4}]
```

$50400x^2y^3$

$\partial_{\{x,3\},\{y,4\}}$**f[x,y]**

$50400x^2y^3$

Derivative 命令也可以用来构造偏导数.假设 f 为 k 元函数,变量为 $x_1, x_2, \cdots, x_k$.

- **Derivative[**n_1, n_2, …, n_k**][f]**给出偏导数 $\dfrac{\partial^n f[x_1,x_2,\cdots,x_k]}{\partial x_1^{n_1}\partial x_2^{n_2}\cdots\partial x_k^{n_k}}$,其中 $n_1+n_2+\cdots+n_k=n$.

例 3

```
f[x_, y_] = x^5y^7;
g = Derivative[3, 4][f];
g[x,y]
```

$50400x^2y^3$

虽然 D 命令可以用来计算偏导数在给定点的值,但 Derivative 命令可能更方便一些.

例 4　令 $f(x,y)=x^3\sin y$，计算在 $(2,\pi)$ 点 f_{xy} 的值.

```
f[x_, y_] = x^3 Sin[y];
D[f[x, y], x, y]/.{x→2, y→π}
-12
Derivativep[1,1][f][2,π]
-12
```

习题解答

10.1　计算函数 $f(x,\ y)=xe^{xy}$ 的各种 1 阶偏导数与 2 阶偏导数.

解 ☞

```
f[x_, y_] = x Exp[x y];
D[f[x, y], x]
e^(xy) + e^(xy) xy
D[f[x, y], y]
e^(xy) x^2
D[f[x, y], {x, 2}]
2 e^(xy) y + e^(xy) xy^2
D[f[x, y], {y, 2}]
e^(xy) x^3
D[f[x, y], x, y]
2 e^(xy) x + e^(xy) x^2 y
```

10.2　$f(x,\ y)$ 的偏导数定义为如下极限：

$$f_x(x,\ y)\ =\ \lim_{h\to 0}\frac{f(x+h,y)-f(x,y)}{h},$$

$$f_y(x,\ y)\ =\ \lim_{h\to 0}\frac{f(x,y+h)-f(x,y)}{h}.$$

利用这个定义计算 $f(x,\ y)=\ln(x^2+y^3)$ 的偏导数，然后再利用 Mathematica 命令 D 进行验证.

解 ☞

```
f[x_, y_] = Log[x^2 + y^3];
Limit[(f[x+h,y] - f[x,y])/h, h→0]
2x/(x^2 + y^3)
D[f[x, y], x]
2x/(x^2 + y^3)
Limit[(f[x,y+h] - f[x,y])/h, h→0]
3y^2/(x^2 + y^3)
D[f[x, y], y]
```

$$\frac{3y^2}{x^2+y^3}$$

10.3 令 $z = e^{xy}$，计算 $\frac{\partial^3 z}{\partial x^2 \partial y}$.

解 ☞

```
z = Exp[x y];
D[z, {x, 2}, y]或者 ∂{x,2},y z
```

$2e^{xy}y + e^{xy}xy^2$

10.4 验证 $u = e^{-a^2k^2t}$ 是热传导方程 $\frac{\partial u}{\partial t} = a^2\frac{\partial^2 u}{\partial x^2}$ 的一个解.

解 ☞

```
Clear[a, k]
u = Exp[-a^2k^2t] Sin[k x];
lhs = D[u, t]
```

$-a^2 e^{-a^2k^2t}k^2\text{Sin}[kx]$

```
rhs = a^2 D[u, {x, 2}]
```

$-a^2 e^{-a^2k^2t}k^2\text{Sin}[kx]$

```
lhs == rhs
True
```

10.5 三元函数 $f(x, y, z)$ 称为调和函数是指它满足拉普拉斯方程：$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = 0$.

令 $f(x,y,z) = \frac{1}{\sqrt{x^2+y^2+z^2}}$，计算它的 f_{xx}，f_{yy} 与 f_{zz}，并证明它是一个调和函数.

解 ☞

```
f[x_, y_, z_] = 1/Sqrt[x^2 + y^2 + z^2];
∂{x,2}f[x, y, z]//Together
```

$$\frac{2x^2-y^2-z^2}{(x^2+y^2+z^2)^{5/2}}$$

```
∂{y,2}f[x, y, z]//Together
```

$$\frac{-x^2+2y^2-z^2}{(x^2+y^2+z^2)^{5/2}}$$

```
∂{z,2}f[x, y, z]//Together
```

$$\frac{-x^2-y^2+2z^2}{(x^2+y^2+z^2)^{5/2}}$$

```
%%% + %% + %//Together
0
```

10.6 由 $z = f(x, y)$ 定义的曲面在 (x_0, y_0, z_0) 点的切平面方程为

$$z = z_0 + f_x(x_0,y_0)(x-x_0) + f_y(x_0,y_0)(y-y_0).$$

确定抛物面 $z = 10 - x^2 - 2y^2$ 在(1, 2)点的切平面方程.画出该抛物面与这个切平面

的图形.

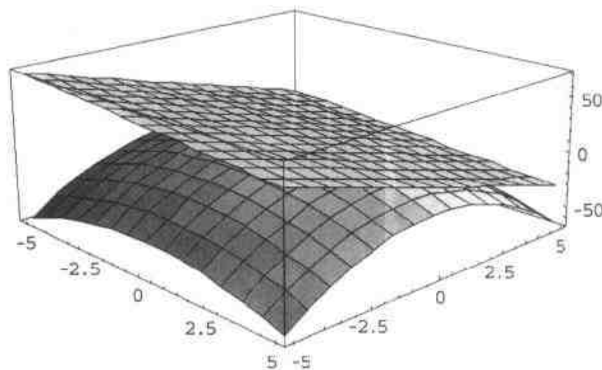
解 ☞

```
f[x_, y_] = 10 - x^2 - 2y^2;
z = f[1, 2] + Derivative[1,0][f][1,2](x - 1)
        + Derivative[0,1][f][1,2](y - 2)//Expand
```

19 - 2x - 8y　　切平面方程为 $z = 19 - 2x - 8y$.

```
g1 = Plot3D[f[x, y], {x, -5, 5}, {y, -5, 5},
    DisplayFunction -> Identity];
g2 = Plot3D[z, {x, -5, 5}, {y, -5, 5},
    DisplayFunction -> Identity];
Show[g1, g2, ViewPoint -> {2.330, -2.223, 1.040},
    DisplayFunciton -> $DisplayFunction];
```

> DisplayFunction → Identity 使得图形结果不显示出来(见第四章).



10.7　曲面 $f(x, y, z) = 0$ 在点 (x_0, y_0, z_0) 的切平面方程为

$$f_x(x_0,y_0,z_0)(x - x_0) + f_y(x_0,y_0,z_0)(y - y_0) + f_z(x_0,y_0,z_0)(z - z_0) = 0.$$

画出球面 $x^2 + y^2 + z^2 = 14$ 与它在点(1, 2, 3)的切平面的图形.

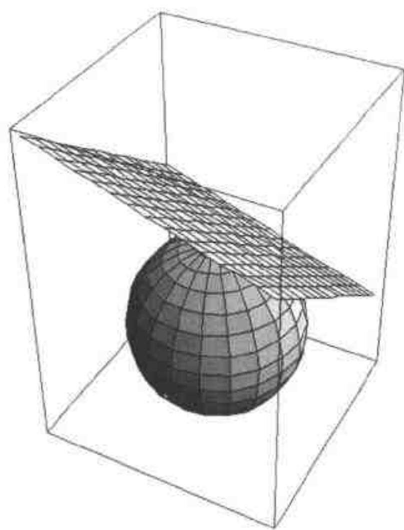
解 ☞

球面中心在原点、半径为 $\sqrt{14}$. 它的方程可写为 $x^2 + y^2 + z^2 - 14 = 0$. 我们可以用图形基本单元 **Sphere** 构造它的图形(见第五章).

```
<<Graphics`Shapes`
f[x_, y_, z_] = x^2 + y^2 + z^2 - 14;
g1 = Graphics3D[Sphere[√14, 20, 15]];
a = Derivative[1, 0, 0][f][1, 2, 3];
b = Derivative[0, 1, 0][f][1, 2, 3];
c = Derivative[0, 0, 1][f][1, 2, 3];
Solve[a(x - 1) + b(y - 2) + c(z - 3) == 0, z]
```

$\{\{z \to \frac{1}{3}(14 - x - 2y)\}\}$

```
g2 = Plot3D[1/3(14 - x - 2y), {x, -5, 5}, {y, -5, 5},
                                    DisplayFunction -> Identity];
Show[g1, g2, DisplayFunction -> $DisplayFunction]
```



10.2　极大值与极小值

我们称函数 $f(x, y)$ 在点(x_0, y_0)达到局部(或相对)极大值是指存在以(x_0, y_0)为中心的开圆盘,对于其中所有点(x, y)都有 $f(x, y)\leqslant f(x_0, y_0)$成立.类似地可以定义局部极小值(不等式反号).如果 f 在(x_0, y_0)点达到局部极大值或者局部极小值,我们统称 f 在(x_0, y_0)点达到局部极值.

对于可微函数 f 而言, f 在(x_0, y_0)点达到局部极值的必要条件是 $f_x(x_0, y_0)=f_y(x_0, y_0)=0$.这样的点称为函数 f 的临界点.

例 5　为了求出 $f(x, y)=x^4+y^4-4xy$ 的临界点,我们计算它的一阶偏导数,然后令它们为零,再求解所得方程组.

```
f[x_, y_] = x^4 + y^4 - 4x y;
pdx = D[f[x,y], x]
4x^3 - 4y
pdy = D[f[x, y], y]
- 4x + 4y^3
Solve[{pdx == 0, pdy == 0}, {x, y}]
{{x→-1, y→-1}, {x→0, y→0},
 {x→-i, y→i}, {x→i, y→-i}, {x→1, y→1},
 {x→-(-1)^(1/4), y→-(-1)^(3/4)}, {x→(-1)^(1/4), y→(-1)^(3/4)},
 {x→-(-1)^(3/4), y→-(-1)^(1/4)}, {x→(-1)^(3/4), y→(-1)^(1/4)}}
```

因此可见,实临界点仅有 $(-1, -1)$, $(0, 0)$与$(1, 1)$.

不巧的是,并不是所有的临界点都必定是极值点.为了确定在一个临界点处是否达到极值,并且如果为极值点的话,它到底是极大值点,还是极小值点,我们需要利用2阶偏导数检测条件:

令 $D(x, y)=f_{xx}(x,y)f_{yy}(x,y)-[f_{xy}(x,y)]^2$,并且$(x_0, y_0)$ 为 f 的一个临界点,

1. 如果 $D(x_0, y_0)>0$ 并且 $f_{xx}(x_0, y_0)>0$,那么(x_0, y_0)点为 f 的局部极大值点.
2. 如果 $D(x_0, y_0)>0$ 并且 $f_{xx}(x_0, y_0)<0$,那么(x_0, y_0)点为 f 的局部极小值点.
3. 如果 $D(x_0, y_0)<0$,那么(x_0, y_0)点既不是 f 的局部极大值点,也不是局部极小值

点.我们称(x_0, y_0)为 f 的鞍点.

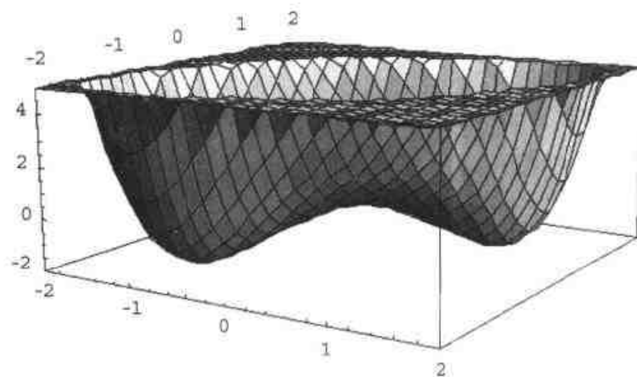
如果 $D(x_0, y_0)=0$,上述检测条件不适用.

例 6 继续考虑前面那个例子.此时先定义 $D(x, y)$.(在下面使用符号 d,以避免与 Mathematica 求导命令 D 混淆.)

```
d[x_, y_] = ∂{x,2}f[x,y]∂{y,2}f[x,y] - (∂x,yf[x,y])^2;
d[0,0]
-16                ←负数,所以(0, 0)点为鞍点.
d[1,1]
128
∂{x,2}f[x, y]/.{x→1, y→1}
12                 ←(1, 1)点为局部极大值点.
d[-1, -1]
128
∂{x,2}f[x,y]/.{x→-1, y→-1}
12                 ←(-1, -1)点为局部极大值点.
```

这时当然值得画出这个函数的图形,以说明上述结果的合理性.利用 Mathematica 可以很容易地做到这一点,虽然为了使细节更清晰,有时可能需要进行几次试验,以确定所需要的选项.

```
Plot3D[f[x,y], {x, -2, 2}, {y, -2, 2}, PlotRange→{-2, 5},
      ViewPoint→{1.761, -2.816, 0.647},PlotPoints→30];
```



为了求出在满足条件 $g(x,y)=0$ 的限制下,$f(x, y)$ 的最(极)大值与最(极)小值,可以利用拉格朗日乘数法.从几何意义上看,可以证明在 f 的极大值(极小值)所达到的点上,f 的层次曲线与 g 的层次曲线具有公共切线.在该点上,f 的梯度与 g 的梯度是平行的,即 $\nabla f(x, y)=\lambda\nabla g(x, y)$.由此可得:

$$f_x(x,y) = \lambda g_x(x, y),$$
$$f_y(x, y) = \lambda g_y(x, y).$$

利用这两个方程,再加上 $g(x,y)=0$,λ 就可以被消去,并可以求出确定 f 的极大值与极小值所对应的 x 与 y 的值.下面的例子演示了这个步骤.

例 7 假设要计算在满足条件 $x^2+y^2=4$ 的前提下,$f(x, y)=2x^2+3y^2$ 的最大值与最小值.为此定义 $g(x, y)=x^2+y^2-4$,并利用下述步骤消去参变量 λ.

```
f[x_, y_] = 2x^2 + 3y^2; g[x_, y_] = x^2 + y^2 - 4;
   conditions = Eliminate[{∂xf[x,y] == λ∂xg[x,y],
```

```
                                      ∂yf[x,y] == λ∂yg[x,y], g[x, y] == 0},λ]
x^2 == 4 - y^2 && xy == 0 && - 4y + y^2 == 0
pts = Solve[conditions]
{{x→ - 2, y→0}, {x→0, y→ - 2}, {x→0, y→2}, {x→2, y→0}}
```

> Eliminate[方程组,变量]从一联立方程组中消去指定的变量.

为了确定 f 的最大值与最小值,我们计算 f 在这些点的值.

```
functionvalues = f[x,y]/.pts
{8, 12, 12, 8}
Max[functionvalues]
12
Min[functionvalues]
8
```

拉格朗日乘数法可以推广到有三个(或多个)变量的函数上.

例 8 为了求出函数 $f(x, y, z)=xyz$ 在满足条件 $x^2+2y^2+3z^2=6$ 的前提下的最大值与最小值,我们定义 $g(x, y, z)=x^2+2y^2+3z^2-6$.

```
f[x_, y_, z_] = x y z;
g[x_, y_, z_] = x^2 + 2y^2 + 3z^2 - 6;
conditions = Eliminate[{∂xf[x,y,z] == λ∂xg[x,y,z],
∂yf[x,y,z] == λ∂yg[x,y,z], ∂zf[x,y,z] == λ∂zg[x,y,z],
g[x,y,z] == 0},λ]
x^2 == 6 - 2y^2 - 3z^2 && 4y^2z == z(6 - 3z^2) &&
x(2y^2 - 3z^2) == 0 && xy( - 2 + 3z^2) == 0 && x z( - 2 + 3z^2) == 0 &&
y z( - 2 + 3z^2) == 0 && 4z - 8z^3 + 3z^5 == 0 && y^3 + y( - 3 + 3z^2) == 0
pts = Solve[conditions]
```

$\{\{x\to0, y\to0, z\to-\sqrt{2}\}, \{x\to0, y\to0, z\to-\sqrt{2}\},$
$\{x\to0, y\to0, z\to\sqrt{2}\}, \{x\to0, y\to0, z\to\sqrt{2}\},$
$\{x\to0, y\to-\sqrt{3}, z\to0\}, \{x\to0, y\to-\sqrt{3}, z\to0\},$
$\{x\to0, y\to\sqrt{3}, z\to0\}, \{x\to0, y\to\sqrt{3}, z\to0\},$
$\{x\to-\sqrt{2}, y\to-1, z\to-\sqrt{\frac{2}{3}}\}, \{x\to-\sqrt{2}, y\to-1, z\to\sqrt{\frac{2}{3}}\},$
$\{x\to-\sqrt{2}, y\to1, z\to-\sqrt{\frac{2}{3}}\}, \{x\to-\sqrt{2}, y\to1, z\to\sqrt{\frac{2}{3}}\},$
$\{x\to\sqrt{2}, y\to-1, z\to-\sqrt{\frac{2}{3}}\}, \{x\to\sqrt{2}, y\to-1, z\to\sqrt{\frac{2}{3}}\},$
$\{x\to\sqrt{2}, y\to1, z\to-\sqrt{\frac{2}{3}}\}, \{x\to\sqrt{2}, y\to1, z\to\sqrt{\frac{2}{3}}\},$
$\{x\to-\sqrt{6}, y\to0, z\to0\}, \{x\to-\sqrt{6}, y\to0, z\to0\}$
$\{x\to\sqrt{6}, y\to0, z\to0\}, \{x\to\sqrt{6}, y\to0, z\to0\}\}$

```
functionvalues = f[x,y,z]/.pts
```

$\{0,0,0,0,0,0,0,0, -\frac{2}{\sqrt{3}}, \frac{2}{\sqrt{3}}, \frac{2}{\sqrt{3}}, -\frac{2}{\sqrt{3}}, \frac{2}{\sqrt{3}}, -\frac{2}{\sqrt{3}}, -\frac{2}{\sqrt{3}}, \frac{2}{\sqrt{3}}, 0,0,0,0\}$

```
Min[functionvalues]
```

$-\frac{2}{\sqrt{3}}$

```
Max[functionvalues]
```

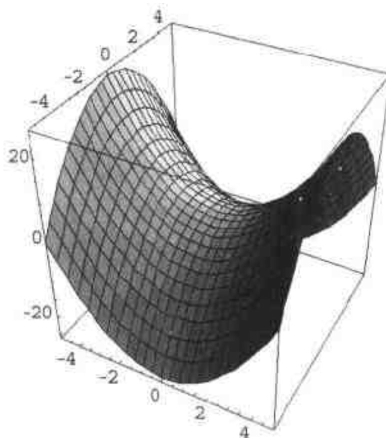
$\frac{2}{\sqrt{3}}$

习题解答

10.8　求出函数 $f(x, y)=x^2-y^2$ 的所有相对极值,并画出它的图形.

解 ☞

```
f[x_, y_] = x^2 - y^2;
Solve[{∂x f[x, y] == 0, ∂y f[x, y] == 0}, {x, y}]
{{x→0, y→0}}
d[x_, y_] = ∂{x,2} f[x, y] ∂{y,2} f[x, y] - (∂x,y f[x, y])^2;
d[0, 0]
-4
Plot3D[f[x, y], {x, -5, 5}, {y, -5, 5},
BoxRatios →{1, 1, 1}, PlotPoints →25];
```



10.9　求出 $f(x, y)=xye^{-x^2-y^2}$ 的所有相对极值点,并画出它的图形.

解 ☞

```
f[x_, y_] = x y Exp[-x^2 - y^2];
pdx = ∂x f[x, y]//Factor
```

$-e^{-x^2-y^2}(-1+2x^2)y$

```
pdy = ∂y f[x, y]//Factor
```

$-e^{-x^2-y^2}x(-1+2y^2)$

如果这时就用 Solve 求解出使所有偏导数都等于 0 的点,那么会得到一条出错信息,因为在方程组中包含了(非代数的)指数.然而,由于 $e^{-x^2-y^2}$ 不可能等于零,因此我们可以忽略它.

```
Solve[{(-1 + 2x^2) y == 0, x(-1 + 2y^2) == 0}, {x, y}]
```

$\{\{x\to 0, y\to 0\}, \{x\to -\frac{1}{\sqrt{2}}, y\to -\frac{1}{\sqrt{2}}\},$
$\{x\to -\frac{1}{\sqrt{2}}, y\to \frac{1}{\sqrt{2}}\}, \{x\to \frac{1}{\sqrt{2}}, y\to -\frac{1}{\sqrt{2}}\}, \{x\to \frac{1}{\sqrt{2}}, y\to \frac{1}{\sqrt{2}}\}\}$

```
d[x_, y_] = ∂{x,2}f[x, y]∂{y,2}f[x, y] - (∂x,yf[x, y])^2;
d[0,0]
```

-1　　←负数，不是相对极值.

```
d[-1/√2, -1/√2]
```

$\frac{4}{e^2}$

```
∂{x,2}f[x,y]/.{x → -1/√2, y → -1/√2}
```

$-\frac{2}{e}$　　←$\left(\frac{-1}{\sqrt{2}},\frac{-1}{\sqrt{2}}\right)$为相对极大值点.

```
d[-1/√2, 1/√2]
```

$\frac{4}{e^2}$

```
∂{x,2}f[x,y]/.{x → -1/√2, y → 1/√2}
```

$\frac{2}{e}$　　←$\left(\frac{-1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$为相对极小值点.

```
d[1/√2, -1/√2]
```

$\frac{4}{e^2}$

```
∂{x,2}f[x,y]/. {x → 1/√2, y → -1/√2}
```

$\frac{2}{e}$　　←$\left(\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}}\right)$为相对极小值点.

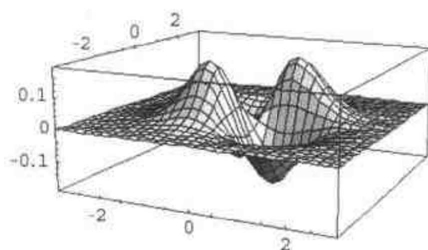
```
d[1/√2, 1/√2]
```

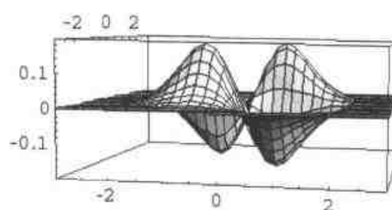
$\frac{4}{e^2}$

```
∂{x,2}f[x,y]/. {x → 1/√2, y → 1/√2}
```

$\frac{2}{e}$　　←$\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$为相对极大值点.

我们用两个视图显示这个函数的图形.





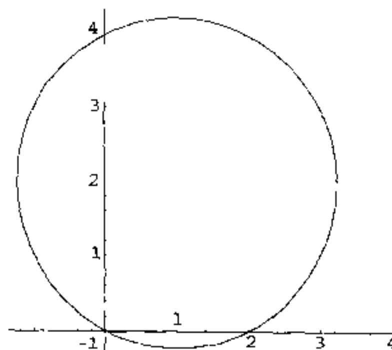

```
Plot3D[f[x, y], {x, -3, 3}, {y, -3, 3}, PlotPoints → 30,
    ViewPoint → {1.391, -3.001, 0.713}, PlotRange → All];
Plot3D[f[x, y], {x, -3, 3}, {y, -3, 3}, PlotPoints → 30,
    ViewPoint → {0.617, -3.318, 0.245}, PlotRange → All];
```

10.10　利用拉格朗日乘数法求出在圆 $x^2+y^2-2x-4y=0$ 上离 $P(4, 4)$ 最近与最远的点.

解 ☞

```
<<Graphics`ImplicitPlot`
circ = ImplicitPlot[x^2 + y^2 - 2x - 4y == 0, {x, -5, 5},
```

```
                    DisplayFunction → Identity];
p = Graphics[{PointSize[.02], Point[{4, 4}]}];
Show[circ, p, DisplayFunction → $ DisplayFunction];
```




```
f[x_, y_] = (x - 4)^2 + (y - 4)^2;        ←我们要计算到 P 点距离平方的极值.
g[x_, y_] = x^2 + y^2 - 2x - 4y;
conditions = Eliminate[{∂xf[x, y] == λ∂xg[x, y],
∂yf[x, y] == λ∂yg[x, y], g[x, y] == 0}, λ]
```

$2x == -4 + 3y$ && $-52y + 13y^2 == -32$

```
pts = Solve[conditions]
```

$\left\{\left\{x \to \frac{1}{13}(13 - 3\sqrt{65}), y \to \frac{2}{13}(13 - \sqrt{65})\right\},\right.$

$\left.\left\{x \to \frac{1}{13}(13 + 3\sqrt{65}), y \to \frac{2}{13}(13 + \sqrt{65})\right\}\right\}$

```
√f[x, y]/.pts //N
```

{5.84162, 1.36948}

根据所得的距离，可知其第一个是离 P 最远的点，第二个是离 P 最近的点.

10.11 求出球面 $x^2 + y^2 + z^2 = 1$ 上离(1, 2, 3)点最近与最远的点.

解 ☞

```
f[x_, y_, z_] = (x - 1)^2 + (y - 2)^2 + (z - 3)^2;
g[x_, y_, z_] = x^2 + y^2 + z^2 - 1;
conditions = Eliminate[{∂xf[x, y, z] == λ∂xg[x, y, z],
  ∂yf[x, y, z] == λ∂yg[x, y, z], ∂zf[x, y, z] == λ∂zg[x, y, z], g[x, y, z] == 0}, λ]
```

$3x == z$ && $3y == 2z$ && $14z^2 == 9$

```
pts = Solve[conditions, {x, y, z}]
```

$\left\{\left\{x \to -\frac{1}{\sqrt{14}}, y \to -\sqrt{\frac{2}{7}}, z \to -\frac{3}{\sqrt{14}}\right\},\right.$　　←最远点.

$\left.\left\{x \to \frac{1}{\sqrt{14}}, y \to \sqrt{\frac{2}{7}}, z \to \frac{3}{\sqrt{14}}\right\}\right\}$　　←最近点.

```
√f[x, y, z]/.pts//N
```

{4.74166, 2.74166}

10.3　全微分

在 10.1 节介绍的命令 D 可以计算多元函数的偏导数，在微分中所有不同于指定变量的其它变量都认为是常数．如果 f 为一个函数，不妨假设它是二元函数，两个变量分别为 x 与 y，但 y 仍是 x 的函数，那么 D 就会给出不正确的答案．

命令 **Dt** 返回函数的全微分．

- **Dt[f[x,y]]** 给出函数 f[x,y]的全微分．
- **Dt[f[x,y],x]**给出函数 f[x,y]相对于变量 x 的全微分．

当然，f 为二元函数或更多元函数，其中的独立变量可以是定义 f 的任意一个变量，在上面的描述中用 x 表示．

例 9
```
f[x_, y_] = x^2 y^3;
Dt[f[x,y]]
2xy^3 Dt[x] + 3x^2 y^2 Dt[y]
Dt[f[x,y],x]
2xy^3 + 3x^2 y^2 Dt[y,x]
Dt[f[x,y], y]
3x^2 y^2 + 2xy^3 Dt[x,y]
```

如果 $z=f(x,\ y)$，其中 y 为 x 的函数，那么 **D[f[x,y],x]**给出$\dfrac{\partial z}{\partial x}$，而 **Dt[f[x,y], x]**给出$\dfrac{dz}{dx}$．

例 10　假设 $f(x,\ y)=x^4y^5$，其中 $y=x^3$．下面的命令序列给出不正确的求导结果．
```
z = x^4 y^5;
D[z,x]/.y -> x^3
4x^18
```
实际上，由于 $z=x^{19}$，所以 dz/dx 应等于 $19x^{18}$．
```
Dt[z,x]/. y -> x^3
19x^18
```

在有些表达式中，用字母表示的常数可能会导致混淆．选项 **Constants** 可以告诉 Mathematica 把一个特定的符号看作常数处理．

- **Constants →对象列表**　使得所有出现在对象列表中的符号被当做常数处理．

例 11
```
Dt[x^n, x]
x^n (n/x + Dt[n,x] Log[x])
Dt[x^n, x, Constants -> {n}]
n x^(-1+n)
```

习 题 解 答

10.12　令 $z=\sin xy$，$x=1$，$y=2$，$dx=\Delta x=0.03$，$dy=\Delta y=0.02$．计算 dz 的值，并与 Δz 的值进行比较．

解　☞
```
z = f[x_, y_] = Sin[x y];
Δz = f[x + Δx, y + Δy] - f[x, y];
```

```
Dt[z]/.{x→1, y→2, Dt[x]→0.03, Dt[y]→0.02}
-0.0332917
Δz/.{x→1, y→2, Dt[x]→0.03, Dt[y]→0.02}
-0.0364571
```

10.13 利用微分计算 $e^{0.1}\sqrt{4.01}$的近似值,并确定估计值的相对误差.

解 ☞

```
f[x_, y_]=Exp[x] Sqrt[y];
approximation=f[0, 4]+Dt[f[x,y]]/.{x→0, y→2, Dt[x]→0.1, Dt[y]→0.01}
2.2025
exactvalue=f[0.1, 4.01]
2.2131
percenterror=Abs[approximation-exactvalue]/exactvalue *100;
Print["The error is", percenterror, "%"]
The error is 0.479103%
```

10.14 有一个锡杯,高 8cm,半径为 2cm,壁厚 0.03cm,顶部与底部厚度都是 0.02cm,利用微分近似计算构成这个锡杯的金属总体积.

解 ☞

> 高度的改变是顶部与底部改变的总和

```
v=πr^2 h;
Dt[v]/.{h→0, r→2, Dt[h]→0.04, Dt[r]→0.03}
3.51858
```

金属总体积约为 3.52cm³.

10.15 三个电阻器的电阻分别为 R_1, R_2 与 R_3 欧姆,把它们并联起来,那么总的电阻等效于$\frac{1}{\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}}$欧姆.具体地说,如果三个电阻器的电阻分别为 20 欧姆,30 欧姆和 50 欧姆,每个的最大误差分别为 5%,2%和 1%,把它们并联起来,那么组合后电阻的变化范围是多少?

解 ☞

$$f[R1_, R2_, R3_]=\frac{1}{\frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3}};$$

```
Dt[f[R1, R2, R3]
```

$$-\frac{-\frac{Dt[R1]}{R1^2}-\frac{Dt[R2]}{R2^2}-\frac{Dt[R3]}{R3^2}}{\left(\frac{1}{R1}+\frac{1}{R2}+\frac{1}{R3}\right)^2}$$

```
f[20, 30, 50]//N
9.67742
Dt[f[R1, R2, R3]]/.{R1→20, R2→30, R3→50,
    Dt[R1]→1.0, Dt[R2]→0.6, Dt[R3]→0.5}
0.315297
```

总的电阻范围为 9.67742±0.315297 欧姆.

10.4 多重积分

多重积分,或者更准确地说为重复积分,可以用 **Integrate** 命令实现,这实际上是一元积分命令的推广.

- **Integrate[f[x, y], {x, xmin, xmax}, {y, ymin, ymax}]** 计算双重积分 $\int_{xmin}^{xmax}\int_{ymin}^{ymax} f(x,y)\mathrm{d}y\mathrm{d}x$.
- **Integrate[f[x, y, z], {x, xmin, xmax}, {y, ymin, ymax}, {z, zmin, zmax}]**计算三重积分 $\int_{xmin}^{xmax}\int_{ymin}^{ymax}\int_{zmin}^{zmax} f(x,y,z)dzdydx$.

类似地,可以计算更高阶的重积分.注意在 Integrate 中最右边的变量,是最先被计算的积分变量.

另外,也可以重复使用 **BasicInput** 模板上的积分符号进行重积分计算.

例 12 为了计算积分 $\int_1^2\int_1^x (x+y)\mathrm{d}y\mathrm{d}x$, 输入

```
Integrate[x + y, {x, 1, 2}, {y, 1, x}]
```

$\frac{3}{2}$

显然,如果交换积分变量的顺序,就会得到完全错误的答案.

```
Integrate[x + y, {y, 1, x}, {x, 1, 2}]
```

$-2+\frac{3x}{2}+\frac{x^2}{2}$

如果为了模拟标准的数学记号,那么可以借助于 **BasicInput** 模板.

$\int_1^2\int_1^x (x+y)\,\mathrm{d}y\,\mathrm{d}x$

$\frac{3}{2}$

例 13 为了计算三重积分 $\int_0^2\int_0^x\int_0^{xy} xyz\,\mathrm{d}z\mathrm{d}y\mathrm{d}x$,我们可以采用下面两种方法:

```
Integrate[x y z, {x, 0, 2}, {y, 0, x}, {z, 0, x y}]
```

4

或者

$\int_0^2\int_0^x\int_0^{xy} xyz\,dz\,dy\,dx$

4

如果无法给出所要计算积分的准确值,那么可以进行数值积分.

- **NIntegrate[f[x, y], {x, xmin, xmax}, {y, ymin, ymax}]** 给出二重积分 $\int_{xmin}^{xmax}\int_{ymin}^{ymax} f(x,y)\mathrm{d}y\mathrm{d}x$ 的数值近似.
- **NIntegrate[f[x, y, z], {x, xmin, xmax}, {y, ymin, ymax}, {z, zmin, zmax}]**给出三重积分 $\int_{xmin}^{xmax}\int_{ymin}^{ymax}\int_{zmin}^{zmax} f(x,y,z)\mathrm{d}z\mathrm{d}y\mathrm{d}x$ 的数值近似.

可以类似地计算高阶重积分的数值近似.如果使用的是 **BasicInput** 模板,那么可以使用 N 命令(或者在积分式的右边加上//N)以得到数值近似.适用于一元函数的 **NIntegrate** 命令

的所有选项,都可以用在多重积分中.

例 14
```
Integrate[Exp[x^2 y^2], {x, 0, 1}, {y, 0, 1}]
```
$\frac{1}{2}\sqrt{\pi}\int_0^1 \frac{\text{Erfi}[x]}{x}\,dx$
```
NIntegrate[Exp[x^2 y^2], {x, 0, 1}, {y, 0, 1}]
1.1351
```
$\int_0^1\int_0^1 \text{Exp}[x^2y^2]\,dy\,dx$ //N

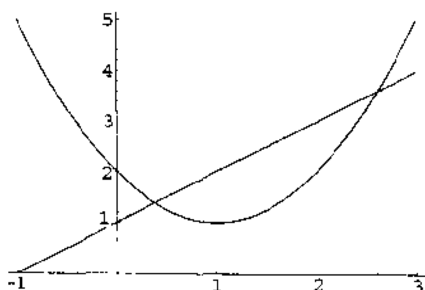
1.1351

习题解答

10.16 利用二重积分计算由抛物线 $y=x^2-2x+2$ 与直线 $y=x+1$ 围成区域的面积.

解 ☞
```
f[x_] = x^2 - 2x + 2;
g[x_] = x + 1;
Plot[{f[x], g[x]}, {x, -1, 3}];
intersections = Solve[f[x] == g[x]]
```
$\left\{\left\{x\to\frac{1}{2}(3-\sqrt{5})\right\},\left\{x\to\frac{1}{2}(3+\sqrt{5})\right\}\right\}$
```
a = intersections[[1,1,2]];
b = intersections[[2,1,2]];
```
$\int_a^b\int_{f[x]}^{g[x]} dy\,dx$

$\frac{5\sqrt{5}}{6}$



在积分前画出所考虑区域的图形,不失为一个好主意.

10.17 给定由抛物线 $y=9-x^2$ 与 x 轴围成的薄板,假设其中每点的密度正比其到 x 轴的距离,在此条件下求出它的质心位置.

解 ☞

用 R 表示由 $y=9-x^2$ 与 x 轴围成的区域.
```
Plot[9 - x^2, {x, -3, 3}];
```
抛物线的图形交 x 轴于 -3 与 3 两点. 则薄板的质心坐标为 $(\bar{x},\bar{y})=\left(\frac{M_y}{M},\frac{M_x}{M}\right)$,其中

$$M_y = \text{相对于 } y \text{ 轴的力矩} = \iint_R x\rho(x,y)\mathrm{d}A,$$

$$M_x = \text{相对于 } x \text{ 轴的力矩} = \iint_R y\rho(x,y)\mathrm{d}A,$$

$$M = \text{薄板的质量} = \iint_R \rho(x,y)\mathrm{d}A,$$

ρ[x_, y_] = k y;

my = $\int_{-3}^{3}\int_{0}^{9-x^2} x\rho[x,y]\,dy\,dx$

0

mx = $\int_{-3}^{3}\int_{0}^{9-x^2} y\,\rho[x,y]\,dy\,dx$

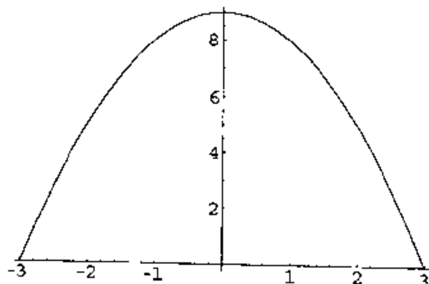
$\frac{23328k}{35}$

m = $\int_{-3}^{3}\int_{0}^{9-x^2} \rho[x,y]\,dy\,dx$

$\frac{648k}{5}$

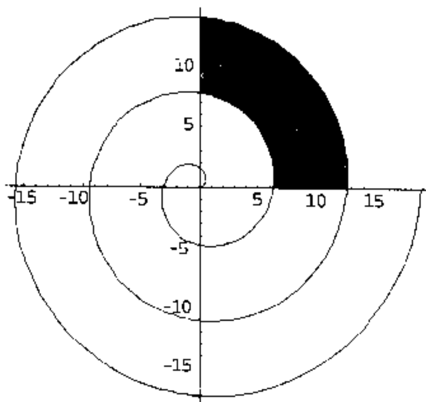
$\left\{\frac{my}{m}, \frac{mx}{m}\right\}$

$\left\{0, \frac{36}{7}\right\}$



10.18　计算阴影部分的面积.这里所给出的曲线为阿基米德螺线,其极坐标方程为 $r=\theta$.显示图形相应于 $0\leqslant\theta\leqslant 6\pi$.

解



已知由极坐标方程界定的区域 R 的面积为 $\iint_R r\mathrm{d}r\mathrm{d}\theta$.给定阴影部分小弧的方程为 $r=\theta, 2\pi\leqslant\theta\leqslant 5/2\pi$,而大弧的方程为 $r=\theta+2\pi, 2\pi\leqslant\theta\leqslant 5/2\pi$.因此它们所包围的区域面积为 $\int_{2\pi}^{5\pi/2}\int_{\theta}^{\theta+2\pi} r\mathrm{d}r\mathrm{d}\theta$.

$\int_{2\pi}^{5\pi/2}\int_{0}^{\theta+2\pi} \mathbf{r\,dr\,d\theta}$

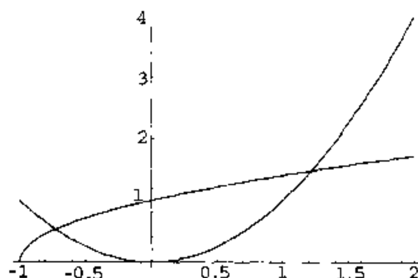
$$\frac{13\pi^3}{4}$$

10.19 计算位于抛物面 $z=x^2+y^2$ 的下面,在 $x-y$ 平面上由 $y=x^2$ 与 $y=\sqrt{x+1}$ 围成区域的上面的立体的体积.

解 ☞

若给定立体由曲面 $z=f(x,\ y)$ 及 x-y 平面围成,在区域 R 的上面,则其体积为 $\iint\limits_R f(x,y)\mathrm{d}A$. 为此,我们首先查看一下 R 的图形.

Plot[{x², √(x+1)}, {x, -1, 2}];



下面计算交点.由于解的性质比较复杂,因此我们使用它的数值近似.

NSolve[x² == √(x+1)]

{{x→1.22074}, {x→-0.724492}}

现在可以把所求体积表示为二重积分.下面给出两种解法.

$$\int_{-.724492}^{1.22074}\int_{x^2}^{\sqrt{x+1}}(x^2+y^2)\,\mathrm{d}y\,\mathrm{d}x\text{//N}$$

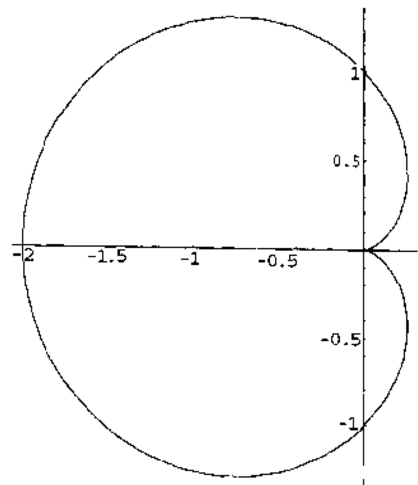
1.11738

NIntegrate[x^2 + y^2, {x, -0.724492, 1.22074}, {y, x^2, Sqrt[x+1]}]

1.11738

10.20 计算在半球面 $z=4-x^2-y^2$ 的下面,x-y 平面上由心脏线 $r=1-\cos\theta$ 围成区域的上面的立体的体积.

解 ☞



下面把问题转化到柱面坐标系中.由于 $r^2=x^2+y^2$,所以半球面的方程变为 $z=4-$

r^2. 积分区域就是由如下所示的心脏线围成的.

```
<<Graphics`Graphics`
PolarPlot[1 - Cos[θ], {θ,0, 2π}];
```

体积 $V = \iint_R (4 - r^2)\mathrm{d}A = \int_0^{2\pi}\int_0^{1-\cos\theta}(4 - r^2)r\mathrm{d}r\mathrm{d}\theta$.

$$\int_0^{2\pi}\int_0^{1-\cos[\theta]}(4 - r^2)r\,dr\,d\theta$$

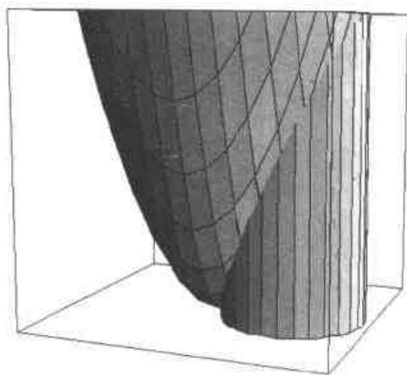
$$\frac{61\pi}{16}$$

10.21 求出位于抛物面 $z = x^2 + y^2$ 的下面, x-y 平面的上面,但包含在圆柱面 $(x - 1)^2 + y^2 = 1$ 的内部的立体的体积.

解

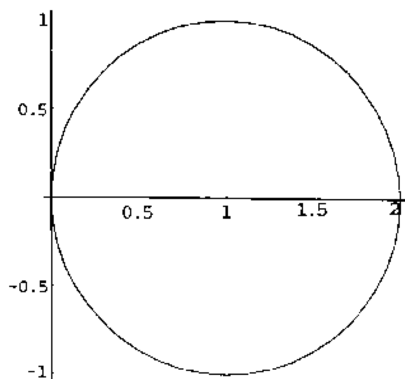
$(x - 1)^2 + y^2 = 1$ 是半径为 1 的圆柱面,其轴线是把 z 轴按向量(1, 0, 0)进行平移得到的.

```
<<Graphics`Shapes`
s1 = Graphics3D[TranslateShape[Cylinder[1, 4, 30], {1, 0, 0}]];
s2 = Plot3D[x^2 + y^2, {x, - 2,2}, {y, - 2,2}, DisplayFunction →Identity];
Show [s1, s2, PlotRange →{0, 4}, ViewPoint →{1.217, - 3.125, 0.447},
    DisplayFunction→ $DisplayFunction];
```



现在我们知道区域的形状,我们还必须查看它在 x-y 平面上的投影.

```
<<Graphics`ImplicitPlot`
```



```
ImplicitPlot[(x - 1)^2 + y^2 == 1, {x, 0, 2}];
```

虽然在直角坐标系中也可以解决这个问题,但在柱面坐标系中就更简单.把圆的方程展开,变为 $x^2+y^2=2x$,它在极坐标中就等价于 $r=2\cos\theta$.当 θ 从 $-\pi/2$ 变到 $\pi/2$ 时,就生成了整个圆周.抛物面的方程 $z=x^2+y^2$ 变为 $z=r^2$.这样所要求的体积可以表示为如下二重积分.

$$\int_{-\pi/2}^{\pi/2}\int_0^{2\mathrm{Cos}[\theta]}(r^2)\,r\,dr\,d\theta$$

$$\frac{3\pi}{2}$$

10.22　在 x-y 平面中区域 R 上的曲面 $z=f(x,\ y)$ 的面积为

$$\iint_R\sqrt{[f_x(x,y)]^2+[f_y(x,y)]^2+1}\,\mathrm{d}A.$$

由此公式计算半径为 a 的球面的表面积.

解 ☞

我们只计算在第一卦限中球面的表面积,然后利用对称性可知乘以 8 即得整个球面的表面积.球面的方程为 $x^2+y^2+z^2=a^2$.从中解出 z,得到 $f(x,\ y)=z=\sqrt{a^2-x^2-y^2}$ 即为上半球面的函数表示.

```
f[x_, y_] = √(a^2 - x^2 - y^2);
1 + (∂x f[x,y])^2 + (∂y f[x,y])^2 //Together
```

$$-\frac{a^2}{a^2+x^2+y^2}$$

由于球面在 $x-y$ 平面上的投影是圆心在原点的半径为 a 的圆,因此利用柱面坐标就很方便.用 r^2 代替 x^2+y^2 得到:

$$\iint_R\sqrt{1+(\partial_x f[x,y])^2+(\partial_y f[x,y])^2}\,\mathrm{d}A=8\int_0^{\pi/2}\int_0^a\sqrt{\frac{a^2}{a^2-r^2}}\,r\,\mathrm{d}r\,\mathrm{d}\theta.$$

$$8\int_0^{\pi/2}\int_0^a\sqrt{\frac{a^2}{a^2-r^2}}\,d\,r\,d\,\theta$$

$$4a^2\pi$$

10.23　由锥面 $z=3\sqrt{x^2+y^2}$ 与半球面 $x^2+y^2+(z-9)^2=9$ 围成"冰淇淋锥",计算它的体积.

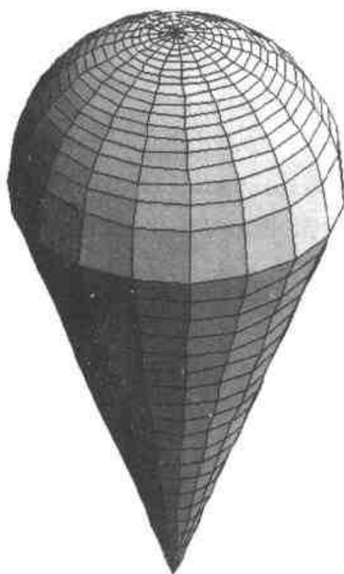
解 ☞

所要计算的体积可以用三重积分 $\iiint_G \mathrm{d}V$ 表示.根据围成立体的曲面性质,利用柱面坐标解决这个问题是非常方便的.首先,把球面方程重写,利用 x, y 表示 z 得到:

```
Solve[x^2 + y^2 + (z - 9)^2 == 9, z]
{{z→9 - √(9 - x^2 - y^2)}, {z→9 + √(9 - x^2 - y^2)}}
```

利用第二个解(对应于上半球面),并用 r^2 代替 x^2+y^2,这样球面的方程变为 $z=9+\sqrt{9-r^2}$.下面画出构成立体的曲面的图形.

```
<<Graphics`ParametricPlot3D`
cone = CylindricalPlot3D[3r, {r, 0, 3}, {θ, 0, 2π},
                                    DisplayFunction →Identity]
hemisphere = CylindricalPlot3D[9 + √(9 - r^2), {r, 0, 3},
```



```
    {θ, 0, 2π},DisplayFunction →Identity];
Show[cone, hemisphere, DisplayFunction → $DisplayFunction, Axes →False,
  Boxed →False];
```

为了计算立体的体积,我们注意区域在 x-y 平面上的投影为一个圆.为了确定它的半径,我们首先计算锥面与球面的交.

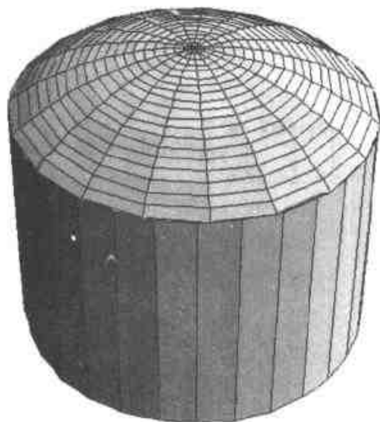
```
Solve[3r == 9 + √(9 - r²)]
{{r→3}}
```

因此区域在 x-y 平面上的投影为半径等于 3 的圆.所要计算的体积为

$$\int_0^{2\pi}\int_0^3\int_{3r}^{9+\sqrt{9-r^2}} r\,dz\,dr\,d\theta$$

45π

10.24　一个筒仓由 x-y 平面上的半径为 3 的直圆柱与半径为 5 的球面构成.计算它的体积.



解

这时仍然在柱面坐标中处理问题是最简单的.球冠的方程为 $z = \sqrt{25 - r^2}$.

```
<<Graphics`Shapes`
```

```
<<Graphics`ParametricPlot3D`
cyl = Graphics3D[Cylinder[3, 4, 30]];
cap = CylindricalPlot3D[ √(25 - r²), {r, 0, 3}, {t, 0, 2π},
     DisplayFunction → Identity];
Show[cyl, cap, PlotRange → {0, 5}, Boxed → False,
  DisplayFunction → $DisplayFunction];
```

立体的投影是半径为 3 的圆,圆心在原点,所以

$$\text{volume} = \int_0^{2\pi}\int_0^3\int_0^{\sqrt{25-r^2}} r\,dz\,dr\,d\theta$$

$$\frac{122\pi}{3}$$

10.25 给定一个半径为 a 的实心半球,其中每点的密度正比于它到 x-y 平面的距离.计算它的质心位置.

解 ☞

质心的坐标$(\bar{x},\bar{y},\bar{z})$满足:

$$\bar{x} = \frac{\iiint\limits_G x\sigma(x,y,z)\mathrm{d}V}{\iiint\limits_G \sigma(x,y,z)\mathrm{d}V},\ \bar{y} = \frac{\iiint\limits_G y\sigma(x,y,z)\mathrm{d}V}{\iiint\limits_G \sigma(x,y,z)\mathrm{d}V},\ \bar{z} = \frac{\iiint\limits_G z\sigma(x,y,z)\mathrm{d}V}{\iiint\limits_G \sigma(x,y,z)\mathrm{d}V}.$$

由题设,密度函数为 $\sigma(x,\ y,\ z) = kz$,其中 k 为比例常数.显然利用球面坐标求解这个问题是最方便的:

$$x = \rho\sin\phi\cos\theta,\ y = \rho\sin\phi\cos\theta,\ z = \rho\cos\theta.$$

```
x = ρSin[φ]Cos[θ];
y = ρSin[φ]Sin[θ];
z = ρCos[φ];
σ = kz;
```

$$\text{mass} = \int_0^{2\pi}\int_0^{\pi/2}\int_0^a \sigma\rho^2\text{Sin}[\phi]\,d\rho\,d\phi\,d\theta$$

$$\frac{1}{4}a^4k\pi$$

$$\text{centerofmass} = \left\{\frac{\int_0^{2\pi}\int_0^{\pi/2}\int_0^a x\sigma\rho^2\text{Sin}[\phi]\,d\rho\,d\phi\,d\theta}{\text{mass}},\ \frac{\int_0^{2\pi}\int_0^{\pi/2}\int_0^a y\sigma\rho^2\text{Sin}[\phi]\,d\rho\,d\phi\,d\theta}{\text{mass}},\ \frac{\int_0^{2\pi}\int_0^{\pi/2}\int_0^a z\sigma\rho^2\text{Sin}[\phi]\,d\rho\,d\phi\,d\theta}{\text{mass}}\right\}$$

$$\left\{0, 0, \frac{8a}{15}\right\}$$

10.26 给定一个半径为 a 的实心半球,其中每点的密度正比于它到底部中心的距离.计算这个实体绕 z 轴的转动惯量.

解 ☞

关于 z 轴的转动惯量为 $\iiint\limits_G [\delta(x,y,z)]^2\sigma(x,y,z)\mathrm{d}V$, 其中 $\delta(x,y,z)$表示点

(x, y, z)到转动轴的距离，$\sigma(x, y, z)$表示(x, y, z)点处的密度．在这个问题中，我们要使用球面坐标系统：

$$x = \rho\sin\phi\cos\theta, y = \rho\sin\phi\cos\theta, z = \rho\cos\theta.$$

$$\delta(x, y, z) = \sqrt{x^2 + y^2} = \sqrt{(\rho\sin\phi\cos\theta)^2 + (\rho\sin\phi\sin\theta)^2} = \rho\sin\phi,$$

$$\sigma(x, y, z) = \sqrt{x^2 + y^2 + z^2} = \rho.$$

$$\int_0^{2\pi}\int_0^{\pi/2}\int_0^{a}(\rho\,\mathrm{Sin}[\phi])^2\rho(\rho^2\mathrm{Sin}[\phi])\,d\rho\,d\phi\,d\theta$$

$$\frac{2a^6\pi}{9}$$

第十一章 常微分方程

11.1 常微分方程的解析解

简单地说,常微分方程就是用于表示函数及其一个或多个导数关系的方程.满足微分方程的函数称为微分方程的一个解.

Mathematica 命令 **DSolve** 用于求解微分方程.同代数或者超越方程一样,要使用双等号==分开方程的两边.

- **DSolve[方程, y[x], x]** 给出独立变量为 x 的微分方程的一般解 y[x].
- **DSolve[方程, y, x]** 给出微分方程的一般解 y,这里把解表示为列表内的纯粹函数(见附录).ReplaceAll (/.)可以用来计算解的值.另外,也可以用 Part 或者[[ ]]从列表中提取解.

例 1 为了求解一阶微分方程$\frac{dy}{dx}=x+y$,只要输入

```
DSolve[y'[x] == x + y[x], y[x], x]
{{y[x]→ -1 - x + e^x C[1]}}
```

例 2 为了用纯粹函数表示$\frac{dy}{dx}=x+y$的解,输入

```
equation = DSolve[y'[x] == x + y[x], y, x]
{{y→(-1 + e^#1 C[1] - #1&)}}
```

如果需要计算解的值,我们可以输入

```
y[x]/.equation
{-1 - x + e^x C[1]}
```

借助于纯粹函数,我们也可以计算解的导数.而如果使用的是例 1 中形式的解,要达到这个目的,就非常麻烦.

```
y'[x]/.equation
{-1 + e^x C[1]}
```

我们可以定义函数 f 用来表示这个微分方程的解.

```
f = equation[[1, 1, 2]]
-1 + e^#1 C[1] - #1 &
```

从而,我们可以直接计算 f 或者其导数的值.

```
f[x]
-1 - x + e^x C[1]
f'[x]
-1 + e^x C[1]
f''[x]
e^x C[1]
```

有一点非常重要,那就是在微分方程中未知函数要表示成 y[x],而不能只用 y.类似地,它的导数要表示成 y'[x],y''[x]等等.下面的例子演示了一些常见的错误.

例 3 **DSolve[y'[x] == x + y, y[x], x]**

$\left\{\left\{y[x] \to \frac{x^2}{2} + xy + c[1]\right\}\right\}$ ← Mathematica 认为 y 为常数.

DSolve[y' == x + y, y, x] ←未知函数及其导数必须用 y[x]与 y'[x]表示.

DSolve::nvld:

The description of the equations appears to be ambiguous or invalid.(方程的描述中包含意义不清或者非法的符号.)

DSolve::deqx:

Supplied equations are not differential equations of the given functions.(所给的不是关于给定函数的微分方程.)

DSolve[y' == x + y, y, x]

没有给定初值条件的微分方程的解中包含一个任意常数,缺省情况下用 C[1]表示.其它的常数(相应于高阶方程)用 C[2],C[3],…表示.如果希望用不同的记号表示常数,可以使用 **DSolveConstants** 选项.

- **DSolveConstants → constantlabel**,指定常数用记号 constantlabel[1], constantlabel[2], …表示.

例 4 **DSolve[y'[x] == x + y[x], y[x], x, DSolveConstants → mylabel]**

{{y[x] → - 1 - x + e^xmylabel[1]}}

高阶微分方程也可以类似求解.导数用 y'[x],y''[x],y'''[x],…表示.另外,也可以用 D,∂ 或者 Derivative 指定导数.

例 5 **DSolve[y''[x] + y[x] == 0, y[x], x]**

{{y[x]→C[2] Cos[x] - C[1] Sin[x]}}

DSolve[D[y[x], {x,2}] + y[x] == 0, y[x], x]

{{y[x]→C[2] Cos[x] - C[1] Sin[x]}}

DSolve[$\partial_{\{x,2\}}$y[x] + y[x] == 0, y[x], x]

{{y[x]→C[2] Cos[x] - C[1] Sin[x]}}

DSolve[Derivative[2][y][x], {x,2}] + y[x] == 0, y[x], x]

{{y[x]→C[2] Cos[x] - C[1] Sin[x]}}

对于复杂的微分方程,如果有可能的话,Mathematica 就会用特殊函数给出其解答.如果 Mathematica 无法求解该方程,它就会返回未求解的方程,或者用未计算出来的积分表示方程的解.对于这些情况,比较适宜的方法是给出方程的数值解(见第 11.2 节).

例 6 $x^2 \frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - 4)y = 0$ 是贝塞耳方程的特殊情形.它的解可以用第一类(BesselJ)和第二类(BesselY)贝塞耳函数表示.

DSolve[x² y''[x] + x y'[x] + (x² - 4)y[x] == 0, y[x], x]

{{y[x]→BesselJ[2, x] C[1] + BesselY[2, x] C[2]}}

例 7 $\frac{d^2y}{dx^2} + \frac{dy}{dx} + y^2 = 0$ 是一个 Mathematica 不能求解的非线性微分方程.

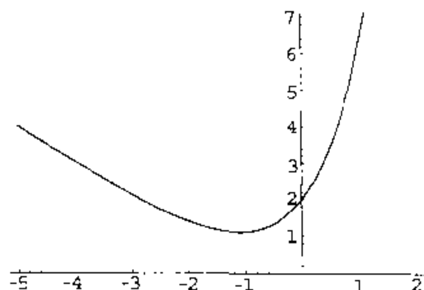
DSolve[y''[x] + y'[x] + y[x]² == 0, y[x], x]

DSolve[y[x]² + y'[x] + y''[x]² == 0, y[x], x]

如果在给定微分方程的同时,也给出了 y 的值,以及有时也给出它的导数值,这时求解 y 的任务就称为初值问题.微分方程与初始条件是在 DSolve 命令中用一个列表来指定的.如果给出的初始条件数目足够,那么就会返回惟一解.

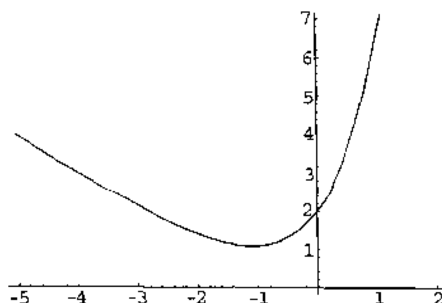
例 8 求解微分方程 $\frac{dy}{dx}=x+y$,初始条件为 $y(0)=2$,然后给出解的图形.

```
equation = DSolve[{y'[x] == x+y[x], y[0]==2}, y[x], x]
{{y[x] → -1 + 3 e^x - x}}
Plot[y[x]/.equation, {x, -5, 2}];
```



下面是另外一种可画出解的图形的解法.

```
equation = DSolve[{y'[x] == x+y[x], y[0] == 2}, y, x]
{{y→(-1 + 3 e^#1 - #1 &)}}
f = equation[[1,1,2]]
-1+3 e^#1 - #1 &
Plot[f[x], {x, -5, 2}];
```



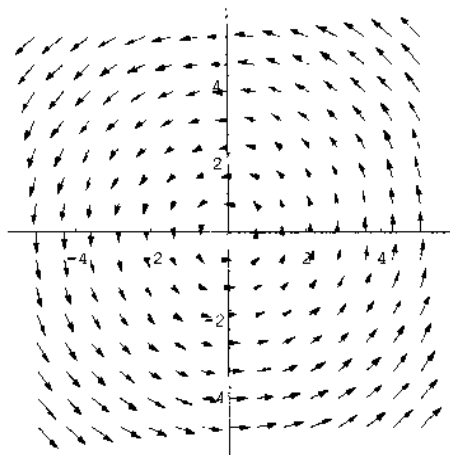
观察一阶微分方程解的性态的实用方法是引进向量场的概念.在$\mathbb{R}^2$中向量场就是一个函数 $\boldsymbol{F}$,在每点(x, y)上都赋予二维向量 $\boldsymbol{F}(x, y)$.通过绘制$\mathbb{R}^2$中向量 $\boldsymbol{F}(x, y)$的有限子集,就可以对 $\boldsymbol{F}$ 的性态有一个几何了解.

- **PlotVectorField[{Fx, Fy}, {x, xmin, xmax}, {y, ymin, ymax}]**生成二维向量场函数 F 的向量场图形,这里 **Fx, Fy** 表示 F 的两个分量.在向量场中点(x, y)处箭头的方向就是该处向量的方向.箭头的梯度正比于向量场的梯度.

PlotVectorField 包含在软件包 Graphics`PlotField`中,因此在使用前要上载这个软件包.

例 9 绘制向量场 $\boldsymbol{F}(x, y)=-y\boldsymbol{i}+x\boldsymbol{j}$ 的图形.在默认情况下,下画出坐标轴,因此下面使用了选项 **Axes →Automatic.**

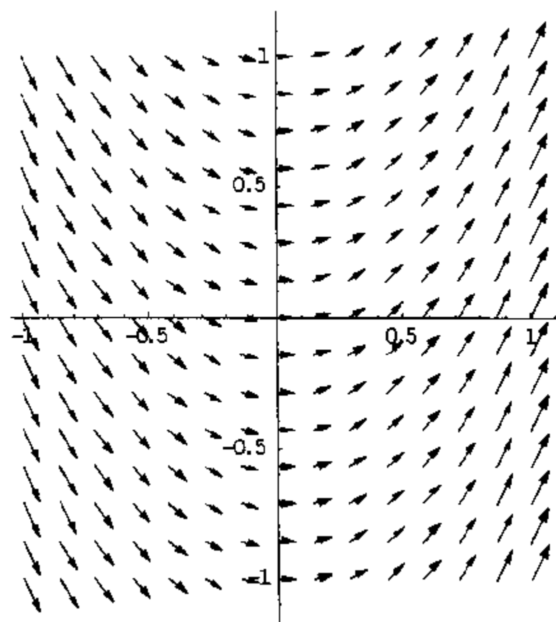
```
<<Graphics`PlotField`
PlotVectorField[{-y, x}, {x, -5, 5}, {y, -5, 5}, Axes →Automatic];
```



任何的一阶微分方程都可以定义一个向量场.实际上,向量场 $\boldsymbol{i}+f(x, y)\boldsymbol{j}$ 对应于微分方程$\dfrac{\mathrm{d}y}{\mathrm{d}x}=f(x,y)$,这栏就生成一个向量场,其中的向量相切于该点的解曲线.下面给出的这个例子,虽然比较简单,但很好地演示了这一点.

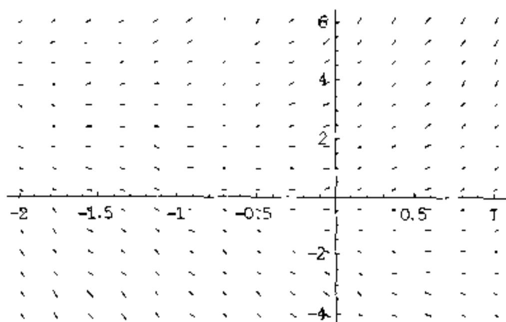
例 10 绘制方程$\dfrac{\mathrm{d}y}{\mathrm{d}x}=2x$ 的解所确定的向量场的图形.方程的解为抛物线 $y=x^2+c$,这一点从下面的图形中可以清楚地看到.

```
<<Graphics`PlotField`
PlotVectorField[{1, 2x}, {x, -1, 1}, {y, -1, 1}, Axes →Automatic];
```



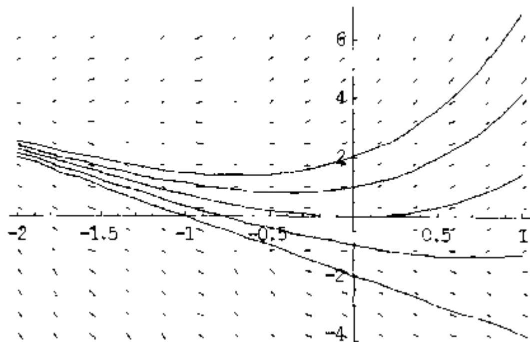
例 11 在这个例子中我们绘制方程$\dfrac{\mathrm{d}y}{\mathrm{d}x}=2x+y$ 的向量场.选项 **HeadLength →0** 使得所得图形显得更清晰.为了比较,我们在向量场中画出初始条件为 $y(0)=$ 2, 1,0,1 与 2 时解的图形.

```
<<Graphics`PlotField`
vf = PlotVectorField[{1, 2x + y}, {x, -2, 1}, {y, -4, 6},
    Axes →Automatic, HeadLength →0,    AspectRatio →1/GoldenRatio,
```



```
  DisplayFunction → $DisplayFunction];
sols = Table[DSolve[{y'[x] == 2x + y[x], y[0] == k}, y[x], x], {k, -2, 2}];
{{{y[x]→ -2 - 2x}}, {{y[x]→ -2 + e^x - 2x}}, {{y[x]→ -2 + 2e^x - 2x}},
 {{y[x]→ -2 + 3e^x - 2x}}, {{y[x]→ -2 + 4e^x - 2x}}}
Do[g[k] =
  Plot[sols[[k, 1, 1, 2]], {x, -2, 1}, PlotRange →All,
  DisplayFunction →Identity], {k, 1, 5}]
Show[g[1], g[2], g[3], g[4], g[5], vf, DisplayFunction → $DisplayFunction];
```

微分方程组是由 n 个微分方程，$n+1$ 个变量构成的方程组. 利用 Mathematica 求解微分方程组的过程与求解单个微分方程是类似的. 下面的例子演示了如何求解微分方程组$\frac{dx}{dt}=t^2, \frac{dy}{dt}=t^3$, 初始条件为 $x(0)=2, y(0)=3$ 的过程.



例 12

```
solution = DSolve[{x'[t] == t^2, y'[t] == t^3, x[0] == 2, y[0] == 3},
  {x[t], y[t]}, t]
```

$\left\{\left\{x[t]\to\frac{1}{3}(6+t^3), y[t]\to\frac{1}{4}(12+t^4)\right\}\right\}$

在给定初始条件时, 也可以不只是指定 f 在某一个点的函数值以及导数值, 而是指定它在两个不同点的值. 这时求解微分方程的问题就成为边值问题. 然而, 与初值问题不同(它对一大类情形都可以证明解是惟一存在的), 即使对于最简单的方程, 边值问题也可能没有解.

例 13 考虑方程$\frac{d^2y}{dx^2}+y=0$, 边值条件为 $y(0)=0, y(\pi)=1$.

```
DSolve[{y''[x] + y == 0, y[0] == 0, y[π] == 1}, y[x], x]
{}
```

对于同样的方程, 如果边值条件改为 $y(0)=0, y(\pi/2)=1$, 那么就存在惟一的解.

```
DSolve[{y''[x] + y == 0, y[0] == 0, y[π/2] == 1}, y[x], x]
{{y[x]→Sin[x]}}
```

习题解答

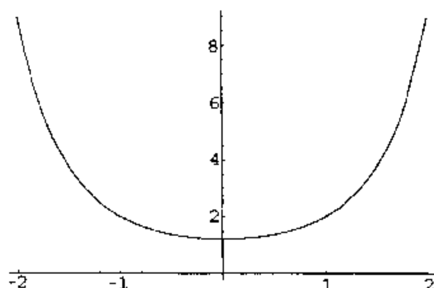
11.1 求解微分方程$\frac{dy}{dx}=xy$，初值条件为$y(1)=2$，并画出解在$-2\leqslant x\leqslant 2$上的图形.

解 ☞

```
equation = DSolve[{y'[x] == x y[x], y[1] == 2}, y[x],x]
```

$\{\{y[x]\to 2e^{-\frac{1}{2}+\frac{x^2}{2}}\}\}$

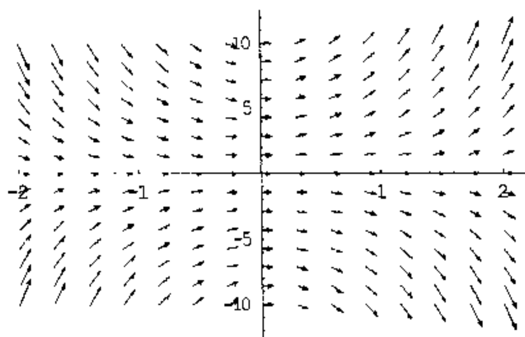
```
Plot[y[x]/.equation, {x, -2, 2}];
```



11.2 画出前一个习题中微分方程所定义的向量场的图形.

解 ☞

```
<<Graphics`PlotField`
PlotVectorField[{1, x y}, {x, -2, 2}, {y, -10, 10},
          AspectRatio →1/GoldenRatio, ScaleFactor →2,
          HeadLength →0.01, Axes →Automatic];
```



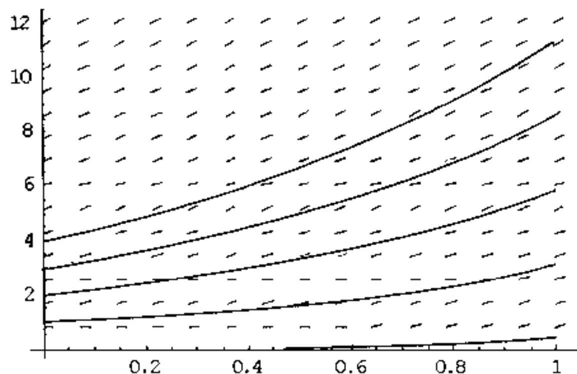
> 选项 **ScaleFactor→长度**对向量进行放缩，从而使最长向量的长度达到指定的长度值. 默认值为 **Automatic**. 对于这个设置，向量长度通常太短，无法看清楚它的方向. **HeadLength** 调整箭头的长度.

11.3 在同一幅图形中画出微分方程$\frac{dy}{dx}=x^2+y$所确定的向量场以及初值条件为$y(0)=0$，1，2，3，4时的图形.

解 ☞

```
<<Graphics`PlotField`
vf = PlotVectorField[{1, x^2 + y}, {x,0,1}, {y,0,12},
        AspectRatio →1/GoldenRatio, ScaleFactor →0.25,
        HeadLength →0.0, Axes →Automatic,
        DisplayFunction →Identity];
sols = Table[DSolve[{y'[x] == x^2 + y[x], y[0] == k}, y[x], x],{k, 0, 4}];
```

```
Do[g[k] = Plot[sols[[k, 1, 1, 2]], {x, 0, 1}, PlotRange → All,
  DisplayFunction → Identity], {k, 1, 5}]
Show[g[1], g[2], g[3], g[4], g[5], vf, DisplayFunction → $DisplayFunction];
```



11.4 所谓逃逸速度就是为了脱离天体重力场作用,而给物体的最低起始速度.计算地球的逃逸速度.

解 ☞

我们可以假设起始速度是沿地球的直径方向的.根据牛顿运动定律,粒子的加速度反比于粒子到地球中心距离的平方.如果 r 表示这个距离,R 表示地球的半径(约等于3960 英里),v 表示粒子的速度,a 表示粒子的加速度,那么 $a=\frac{\mathrm{d}v}{\mathrm{d}t}=\frac{k}{r^2}$.在地球表面($r=R$)上,$a=-g$,其中 $g=32.16\ \mathrm{ft/sec^2}=0.00609\ \mathrm{mi/sec^2}$.由此可知 $k=-gR^2$,所以 $a=-\frac{gR^2}{r^2}$.由于 $a=\frac{\mathrm{d}v}{\mathrm{d}t}$,$v=\frac{\mathrm{d}r}{\mathrm{d}t}$,根据链式法则可知 $a=\frac{\mathrm{d}v}{\mathrm{d}t}=\frac{\mathrm{d}v}{\mathrm{d}r}\frac{\mathrm{d}r}{\mathrm{d}t}=v\frac{\mathrm{d}v}{\mathrm{d}r}$.如果 v_0 表示逃逸速度,就得到微分方程 $v\frac{\mathrm{d}v}{\mathrm{d}r}=-\frac{gR^2}{r^2}$,初值条件为当 $r=R$ 时 $v=v_0$.

```
DSolve[{v[r]v'[r] == - g R^2/r^2, v[R] == v0}, v[r], r]
```

$$\left\{\left\{v[r]\to -\sqrt{-2gR+\frac{2gR^2}{r}+v_0^2}\right\},\left\{v[r]\to \sqrt{-2gR+\frac{2gR^2}{r}+v_0^2}\right\}\right\}$$

由于在地球表面($r=R$)上速度为正的,所以在飞行过程中速度都必须为正的,因此去掉第一个解.而且,$v(r)$一致为正当且仅当 $-2gR+v_0^2\geqslant 0$,即 $v_0\geqslant\sqrt{2gR}$.

```
√2gR/. {g → 0.00609, R → 3960}
6.94498
```

所以逃逸速度为每秒 6.94498 英里.

11.5 根据牛顿散热定律,物体的温度改变速度正比于物体与外界的温度差.如果物体的温度为 70℃,放在温度为 20℃ 的环境中,3 分钟后温度为 40℃,请问 6 分钟后温度是多少?

解 ☞

用 $u(t)$表示物体在时刻 t 的温度,那么$\frac{\mathrm{d}u}{\mathrm{d}t}=k(u-20)$.初值条件为 $u(0)=70$.

```
DSolve[{u'[t] == k(u[t] - 20), u[0] == 70}, u[t], t]
{{u[t] → 20 + 50e^(kt)}}
u[t_] = %[[1, 1, 2]]
20 + 50e^(kt)
```

这时可以用 3 分钟后的温度来确定参数 k 的值.由于在下面是对超越函数应用 Solve 命令,因此尽可以放心地忽略所给出的警告消息.

```
Solve[u[3] == 40, k]
```

Solve::ifun: Inverse functions are being
  used by Solve, so some solutions may not be found.
  (由于 Solve 使用了反函数求解,因此可能有的解没有找到.)

$\left\{\left\{k \to -\frac{1}{3}\text{Log}\left[\frac{5}{2}\right]\right\}\right\}$

```
u[6]/. k → %[[1, 1, 2]]
```

28

6 分钟后的温度为 28℃.

11.6 自由落体下落的加速度为 g,近似等于 32.16 ft/sec^2.如果要考虑空气阻力,那么运动的性态就会发生巨大改变.给定一个物体,其质量为 5 斯拉格,从高为 1000 英尺的地方落下,对于下述两种情况,确定落到地面所需要的时间.(a) 不考虑空气阻力;(b) 假设空气阻力等于物体的速度.

解答

令 $h(t)$表示在时刻 t 时物体的高度,$v(t)$表示此时的速度,$a(t)$表示此时的加速度.注意到 $v(t)=h'(t)$,$a(t)=v'(t)=h''(t)$,由牛顿定律,作用到物体上的外力总和等于质量乘以加速度,即:$ma(t)=\sum F$.在下面我们总是取向上为正向.

(a) 如果不考虑空气阻力,那么物体所受的力只有重力,因此 $ma(t)=-mg$.两边同除以 m,求解微分方程 $h''(t)=-g$,初值条件为 $h'(0)=0$,$h(0)=1000$.

```
g = 32.16;
solution = DSolve[{h''[t] == -g, h'[t] == 0, h[0] == 1000}, h[t], t];
height[t_] = solution[[1,1,2]]
```

$1000. - 16.08t^2$

当物体到达地面时,高度为零.

```
Solve[height[t] == 0, t]
```

{{t → -7.886}, {t → 7.886}}

因此到达地面花费的时间为 7.886 秒.

(b) 如果要考虑空气阻力,那么作用在物体上的外力除重力外,又多了一种,等于 $-v(t)$.微分方程变为

$$ma(t) = -mg - v(t)$$

或

$$mh''(t) = -mg - h'(t)$$

初值条件与 (a) 中相同.

```
m = 5; g = 32.16;
solution = DSolve[{m h''[t] == -mg - h'[t], h'[0] == 0, h[0] == 1000},
                  h[t], t];
height[t_] = solution[[1,1,2]]
```

$1804. - 804.e^{-0.2t} - 160.8t$

```
FindRoot[height[t] == 0, {t, 10}]
```

{{t → 10.6213}}

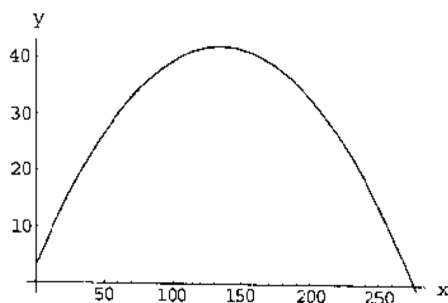
现在需要花费 10.6213 秒到达地面.

11.7　棒球击出的速度为 100ft/sec,与水平线夹角为 30°.球拍离地面 3 英尺.忽略空气与风力的阻力,请问它能否飞过离本垒 200 英尺远、35 英尺高的围墙?

解 ☞

```
g = 32; h = 3; θ = 30 Degree; v0 = 100;
solution = DSolve[{x''[t] == 0, y''[t] == -g,
        x'[0] == v0 Cos[θ], y'[0] == v0 Sin[θ],
        x[0] == 0, y[0] == h}, {x[t], y[t]}, t];
horiz[t_] = solution[[1,1,2]]
50 √3 t
vert[t_] = solution[[1,2,2]]
3 + 50 t - 16 t^2
ParametricPlot[{horiz[t], vert[t]}, {t, 0, 3.2}, AxesLabel → {"x", "y"}];
```

取 $g = 32\text{ft/sec}$




```
Solve[horiz[t] == 200]
{{t → 4/√3}}
vert[t]/. % //N
{33.1367}}
```

从图上可见,当 $x = 200$ 时,是否有 $y \geqslant 35$ 很成问题,因此我们精确地求解出球到达围墙的时间,再计算这时球所在的高度.

由于此时球的高度小于 35 英尺,因此球不会飞过围墙.

11.8　对于习题 11.7 中的棒球,以哪个角度击出,可使得它飞过围墙?

解 ☞

首先求出 y 与 θ 之间的关系.

```
Clear[θ]
g = 32; h = 3; v0 = 100;
solution = DSolve[{x''[t] == 0, y''[t] == -g, x'[0] == v0 Cos[θ],
   y'[0] == v0 Sin[θ], x[0] == 0, y[0] == h}, {x[t], y[t]}, t]
{{x[t] → 100t Sin[θ], y[t] → 3 - 16t^2 + 100t Sin[θ]}}
horiz[t_] = solution[[1, 1, 2]];
vert[t_] = solution[[1,2,2]];
temp = Solve[horiz[t] == 200, t]          ←求解出作为 θ 函数的 t.
{{t → 2 Sec[θ]}}
height[θ_] = vert[t]/.temp                ←定义表示高度的 θ 的函数.
{{3 - 64 Sec[θ]^2 + 200 Tan[θ]}}
NSolve[height[θ Degree] == 35, θ]         ←求出对应于高度为 35 英尺的 θ 值.
Solve::ifun: Inverse functions are          这里的 θ 以角度为单位.
```

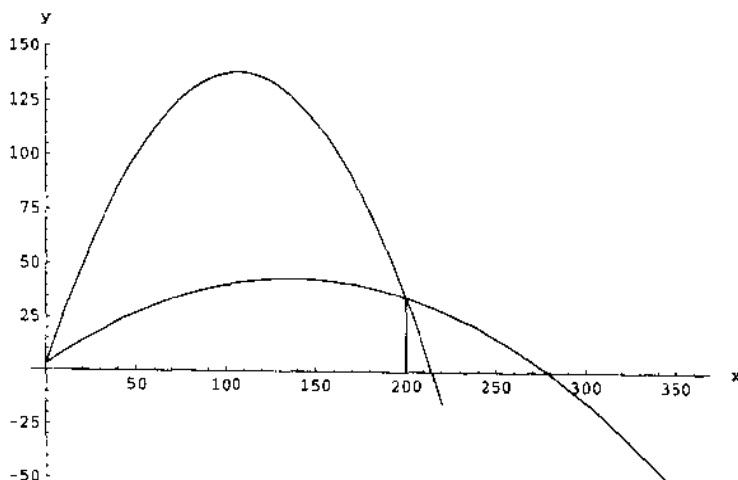
being used by Solve, so some solutions may not be found.

(由于 Solve 使用了反函数求解,从而可能有的解没有找到.)

完全可以放心地忽略这条警告消息.

{{θ→ - 149.364}, {θ→ - 111.545}, {θ→30.6357}, {θ→68.4546}}

这里负值 θ 可以不予考虑.因此只有当 θ 介于 30.6367°与 68.4546°之间时,球才能飞过围墙.为了验证这一结论,我们把这两个角度对应的轨迹画出来.在下图中的竖直线表示离本垒 200 英尺外的 35 英尺高的围墙.




```
θ = 30.6357 Degree;
horiz[t_] = solution[[1, 1, 2]];
vert[t_] = solution[[1,2,2]];
graph1 = ParametricPlot[{horiz[t], vert[t]}, {t, 0, 6},
  DisplayFunction →Identity];
θ = 68.4546 Degree;
horiz[t_] = solution[[1, 1, 2]];
vert[t_] = solution[[1,2,2]];
graph2 = ParametricPlot[{horiz[t], vert[t]}, {t, 0, 6},
  DisplayFunction →Identity];
graph3 = Graphics[Line[{{200, 0}, {200,35}}]];
Show[graph1, graph2, graph3, PlotRange →{ - 50, 150},
  AxesLabel →{"x", "y"}, DisplayFunction → $ DisplayFunction];
```

11.9 有一种细菌,其繁殖速度正比于即时的细菌总数.如果从某时刻开始,一天后有 50 个细菌,两天后有 1 200 个细菌,那么 4 天后有多少个细菌?

解 ☞

这时所满足的微分方程为$\frac{dN}{dt}=kN$,其中 N 为某时刻细菌的数目,k 为题设中所确定的比例常数.初值条件是当 $t=1$ 时,$N=500$.

```
solution = DSolve[{n'[t] == k n[t], n[1] == 500}, n[t], t]
```

$\{\{n[t] \to 500e^{-k+kt}\}\}$

```
population[t_] = solution[[1,1,2]];
Solve[population[2] == 1200, k]
```

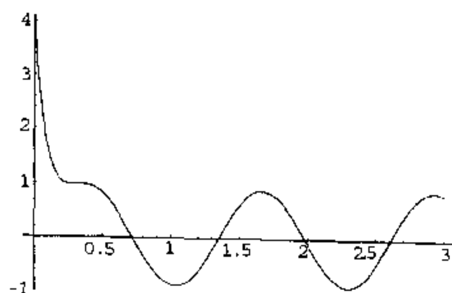
```
Solve::ifun: Inverse functions are being used by Solve, so some solutions may
not be found.
```

(由于 Solve 使用了反函数求解，从而可能有的解没有找到.)

$\left\{\left\{k\to \text{Log}\left[\frac{12}{5}\right]\right\}\right\}$

```
population[4]/. k →Log[12/5]
6912
```

11.10　在一个由电阻与电感组成的简单电路中，施加电动势 E(电压)，那么电流 I 遵从方程：$L\frac{dI}{dt}+RI=E$，其中 E，I 与 L 的单位分别为伏特，安培与亨利. 如果 $R=10$ 欧姆，$L=1$ 亨利，采用交流电源，方程为 $E(t)=10\sin 5t$，初始电流为 4 安培，求出在时刻 t 的电流表达式，并画出前 3 秒的电流图形.



解 ☞

注意　不能用 E 与 I 表示电压与电流.

```
r = 10; l = 1; e[t_] = 10 Sin[5t];
solution = DSolve[{l i'[t] + r i[t] == e[t], i[0] == 4}, i[t], t]
```

$\{\{i[t]\to -\frac{1}{5}e^{-10t}(-22+2e^{10t}\text{Cos}[5t]-4e^{10t}\text{Sin}[5t])\}\}$

```
Plot[solution[[1,1,2]], {t, 0, 3}];
```

11.11　弹簧的一端有质量为 m 的物体，另一端被吊挂起来，这样它就会处于平衡位置不动. 如果把物体向下拖离平衡位置 y_0 距离，并使它具有初速度 v_0，那么它的运动满足微分方程 $m\frac{d^2y}{dx^2}+a\frac{dy}{dt}+ky=0, y'(0)=v_0, y(0)=y_0$. a 是由于摩擦与空气阻力所造成的阻尼常数(由实验来确定)，k 为弹簧遵从胡克定律中的弹性系数. (这里假设向下为正向.)

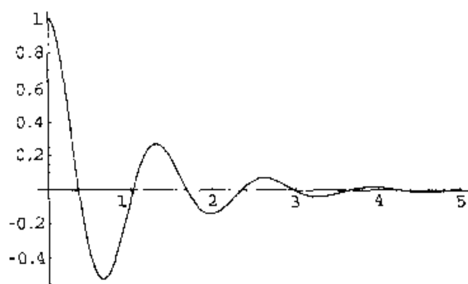
一个质量为 4 斯拉格的物体固定在弹簧的一端，弹簧的弹性系数 k 为 6lb/ft. 物体被向下拖离平衡位置 1ft，然后释放. 假设阻尼常数 $a=1/2$，请确定物体的运动方程，并画出它在前 5 秒的运动图形.

解 ☞

```
m = 1/4; y0 = 1; v0 = 0; a = 1/2; k = 6;
solution = DSolve[{m y''[t] + a y'[t] + k y[t] == 0, y'[0] == v0, y[0] == y0},
  y[t], t]
```

$\left\{\left\{y[t]\to e^{-t}\left(\text{Cos}[\sqrt{23}t]+\frac{\text{Sin}[\sqrt{23}t]}{\sqrt{23}}\right)\right\}\right\}$

```
Plot[solution[[1,1,2]], {t, 0, 5}];
```



11.12 截面均匀的电线被固定在两个电线杆上，这时电线就会下垂，形成悬链线．如果曲线的最低点在 y 轴上，与原点的距离为 y_0，那么可以证明曲线形状所满足的微分方程为 $\frac{d^2y}{dx^2}=\frac{1}{a}\sqrt{1+\left(\frac{dy}{dx}\right)^2}$，$y(0)=y_0, y'(0)=0$，其中 a 为正常数，与电线的物理性质有关．计算悬链线的方程，并画出其图形．

解 ☞

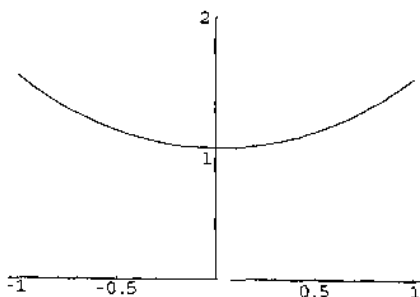
```
solution = DSolve[{y''[x] == 1/a √(1 + y'[x]^2),
y'[0] == 0, y[0] == y0}, y[x], x]
Solve::ifun: Inverse functions are being used by Solve, so some solutions may
not be found.
```

(由于 Solve 使用了反函数求解，从而可能有的解没有找到．)

```
{y[x] → - a + a Cosh[x/a] + y0, y[x] → a + a Cosh[x/a] + y0}
```

由于 $y_0>0$，所以使用第一个解．取 $y_0=a=1$ 以画它的图形．

```
catenary[x_] = solution[[1,2]]/.{a → 1, y0 → 1};
Plot[catenary[x], {x, -1, 1},
Ticks → {Automatic, {0, 1, 2}}, PlotRange → {0, 2}];
```

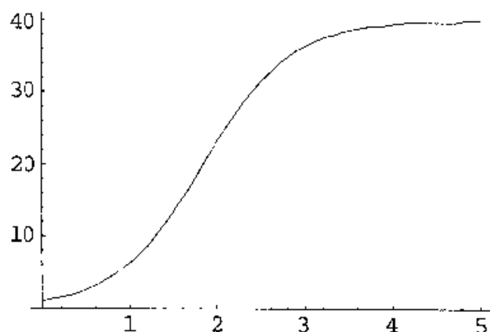


11.13 人口增长的逻辑方程 $\frac{dp}{dt}=ap-bp^2$ 是由生物学家 Pierre Verhulst 在 19 世纪中叶提出的．这里的常数 b 相对于 a 是很小的，从而当人口总数很小时，关于 p 的二次项可以被忽略，人口总数近似于指数增长．然而当人口总数很多时，二次项就会起作用，以减慢人口的增长速度．对于 $a=2, b=0.005$，初始人口为 $p_0=1$(千)时求解出逻辑方程，并确定当 $t\to\infty$ 时人口的极限．

解 ☞

```
solution = DSolve[{p'[t] == a p[t] - b p[t]^2, p[0] == p0}, p[t], t]
```

```
{{p[t_]→ a ⅇ^(at) p0 / (a - b p0 + b ⅇ^(at) p0)}}
population[t_] = solution[[1,1,2]];
Plot[population[t]/.{p0→1,a→2, b→.05}, {t, 0, 5},PlotRange→All];
Limit[population[t]/. {p0→1, a→2, b→.05}, t→∞]
40.
```



11.14　求解边值问题：$\frac{d^2y}{dx^2}+4\pi^2y=0, y(0)=y(1)=0$.

解 ☞

```
DSolve[{y''[x] + 4π² y[x] == 0, y[0] == 0, y[1] == 0}, y[x], x]
{{y[x]→ - C[1]Sin[2πx]}}
```

11.2　常微分方程的数值解

虽然有些微分方程的解可以用初等函数表示出来，但在实际问题中遇到许多微分方程的解做不到这一点. 即使可以证明存在惟一的解，但也只可能给出解的数值近似. **NDSolve** 命令就是为此而设计的.

- **NDSolve[方程, y, {x, xmin, xmax}]**给出在方程中定义的微分方程以及初始条件所确定的解 y 的数值近似，其中独立变量 x 满足 xmin≤x≤xmax.

由于 NDSolve 给出的是微分方程或者微分方程组的一个数值解，因此为了保证解的惟一性，必须指定适当数目的初始条件.

例 14　在这个例子中考虑微分方程$\frac{dy}{dx}=x^2+\sqrt{y}$，初值条件为 $y(0)=1$. 虽然这个方程有惟一解，但这个解不能用 DSolve 求解出它的初等函数表示.

```
DSolve[{y'[x] == x² + √y[x], y[0] == 1}, y[x], x]
DSolve[{y'[x] == x² + √y[x], y[0] == 1}, y[x], x]
```

这时我们只能给出这个方程的数值近似解. 由于数值方法是在有限个点上构造近似值，因此 Mathematica 对近似值进行插值，即构造一个过这些点的光滑函数，并以 InterpolatingFunction 对象的形式返回解.

例 15　

```
temp = NDSolve[{y'[x] == x² + √y[], y[0] == 1}, y, {x, 0, 1}]
{{y→InterpolatingFunction[{{0., 1.}}, <>]}}
```

可以从这个表达式中提取真正的 InterpolatingFunction 对象.

```
solution = temp[[1,1,2]]
InterpolatingFunction[{{0., 1.}}, <>]
```

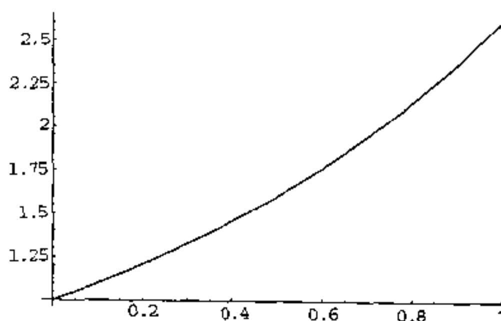
这里只直接显示出 InterpolatingFunction 对象的定义域.其它要素则用< >表示.如果要查看构造中所用的数据,可以输入命令 **FullForm[solution]**.利用这种插值后的解 solution,我们可以计算解在一点或多点上的值,甚至可以画出它的图形.然而这时必须小心处理,保证绘图区间落在 InterpolatingFunction 的定义域内,否则会生成一条警告消息.

```
solution[0.5]
1.60643
solution[1.1]
InterpolatingFunction::dmval:
  Input value {1.1} lies outside the range of data in the interpolating func-
tion. Extrapolation will be used.
```

(输入值{1.1}位于插值函数中数据的外面,因此使用了外插方法.)

```
2.89132        ←从准确度上来看,外插得到的数值是不可靠的.
Table[{x, solution[x]}, {x, 0, 1, .1}]//TableForm
```

0	1.
0.1	1.10284
0.2	1.21273
0.3	1.33181
0.4	1.46228
0.5	1.60643
0.6	1.76656
0.7	1.94504
0.8	2.14429
0.9	2.36672
1.	2.61479




```
Plot[solution[x], {x, 0, 1}];
```

虽然对绝大多数微分方程而言,利用默认设置得到的效果都很好,Mathematica 还是提供了一些选项,以设置相应参数,处理某些反常情形.

- **WorkingPrecision** 指定在内部计算中所用的有效数字位数.缺省情形下,**WorkingPrecision → $MachinePrecision**,通常是 16 位有效数字.
- **PrecisionGoal** 指定在最终答案中的精度位数.缺省情形下,**PrecisionGoal → Automatic**,它等于 WorkingPrecision - 10.
- **AccuracyGoal** **指定准确度位数(即小数点右边的位数).缺省值为 AccuracyGoal →Automatic**,它等于 WorkingPrecision - 10. AccuracyGoal 的值应当设置得大些.如果解很接近零,就要把它设为 Infinity.
- **MaxSteps** 是得到解的最多步数.缺省值为 **MaxSteps →Automatic**,对于常微分方程而言,这个值是 1000.
- **MaxStepSize** 指定在迭代中每一步的最大尺寸.缺省值为 **MaxStepSize →Infinity**.
- **StartingStepSize** 指定起始步长.缺省值为 **StartingStepSize →Automatic**(由 Mathematica 为给定方程自动确定最佳的步长.)

- **InterpolationPrecision**　设置表示解的 InterpolatingFunction 对象的准确度. 缺省值为 **InterpolatingPrecision →Automatic**, 它使用 WorkingPrecision 的值.

例 16　微分方程$\frac{d^2y}{dx^2}+y=0$在初值条件为$y(0)=0, y'(0)=1$时有惟一解$y=\sin x$. 下面计算它在$0\leqslant x\leqslant 1000$时的数值解.

```
equation = NDSolve[{y''[x] + y[x] == 0, y[0] == 0, y'[0] == 1}, y, {x, 0, 1000}].
NDSolve::mxst: Maximum number of 1000
  steps reached at the point x == 199.22439653359802`.
```

（在 x == 199.22439653359802`点时为了得到解,迭代超过了 1000 步.）

```
{{y → InterpolatingFunction[{{0., 199.224}}, <>]}}
```

由于给定的区间[0, 1000]太宽, 在这个区间上为了得到解, 必须超过 1000 步.

```
equation = NDSolve[{y''[x] + y[x] == 0, y[0] == 0, y'[0] == 1}, y, {x, 0, 1000},
      MaxSteps →1000]
{{y → InterpolatingFunction[{{0., 1000.}}, <>]}}
```

已经得到了微分方程的数值解, 我们下面检验它的准确性. 注意准确解在$x=(4k+1)\frac{\pi}{2}$时应等于 1.

```
f = equation[[1,1,2]];
f[π/2]
0.999997
f[633π/2]
0.999407
```

这时离零越远, 误差就变得越大. 为了改变准确度, 我们增大 WorkingPrecision 的值. 这样也自动增大了 PrecisionGoal 与 AccuracyGoal 的值.

```
equation = NDSolve[{y''[x] + y[x] == 0, y[0] == 0, y'[0] == 1}, y, {x, 0,
    1000}, MaxSteps →1000, WorkingPrecision →20]
{{y → InterpolatingFunction[{{0, 1000.0000000000000000}}, <>]}}
f2 = equation[[1,1,2]];
f2[π/2]
0.99999999999317246612
f2[633π/2]
1.000000004055726494
```

这时的计算数值解命令要花费很长时间.

习题解答

11.15　分别用 DSolve 与 NDSolve 求解微分方程$\frac{dy}{dx}=1+\frac{1}{2}y^2, y(0)=1$, 并对结果做一比较.

解

```
equation1 = DSolve[{y'[x] == 1 + 1/2 y[x]^2, y[0] == 1}, y[x], x]
{{y[x]→√2 Tan[1/2 (√2 x + 2ArcTan[1/√2])]}}
solution1[x_] = equation1[[1,1,2]];
```

```
equation2 = NDSolve[{y'[x] == 1 + 1/2 y[x]^2, y[0] == 1}, y[x], {x,0,1}]
{{y[x]→InterpolatingFunction[{{0., 1.}}, <>][x]}}
solution2[x_] = equation2[[1,1,2]];
tabledata = Table[{x, solution1[x], solution2[x]}, {x, 0, 1, .1}];
TableForm[tabledata,TableHeadings→{None,{"x","analytic","numerical"}}]
```

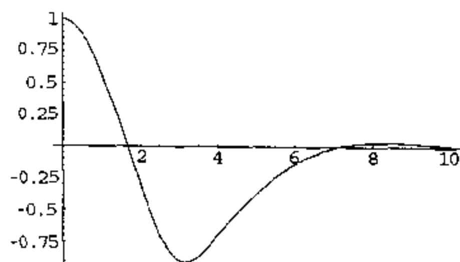
x	analytic	numerical
0	1	1
0.1	1.15817	1.15817
0.2	1.33582	1.33583
0.3	1.53895	1.53896
0.4	1.77601	1.77602
0.5	2.05935	2.05936
0.6	2.40786	2.40788
0.7	2.85196	2.85199
0.8	3.44406	3.44411
0.9	4.28301	4.28309
1.	5.58016	5.58031

从左表可见，数值解与解析解至少在前三位小数上是相同的．当 x 离 0 越来越远时，误差变得越来越大．

11.16 画出微分方程$\frac{d^2y}{dt^2}+\left(\frac{dy}{dt}+1\right)^2\frac{dy}{dt}+y=0, y(0)=1, y'(0)=0$ 在 $0\leqslant t\leqslant 10$ 上解的图形．

解 ☞

```
solution1 = NDSolve[{y''[t] + (y'[t] + 1)^2  y'[t] + y[t] == 0, y[0] ==1,
y'[0] == 0}, y[t], {t, 0, 10}]
{{y[t] → InterpolatingFunction
[{{0.,10.}}, <>][t]}}
Plot[y[t]/.solution1, {t, 0, 10}, PlotRange →All];
```

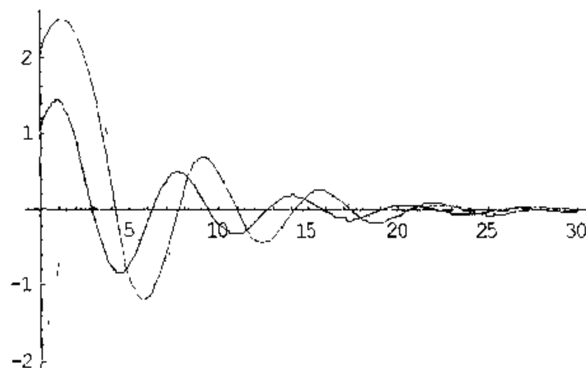


11.17 对于微分方程$\frac{d^2y}{dx^2}+0.3\frac{dy}{dx}+\sin y=0$，分别画出它相应于初值条件为 $y'(0)=1$，$y(0)=-2,-1,0,1,2$ 的五组解在 $0\leqslant x\leqslant 30$ 上的图形．

解 ☞

```
Do[
  {solution =
  NDSolve[{y''[x] + 0.3y'[x] + Sin[y[x]] == 0, y[0] == i,
  y'[0] == 1}, y[x], {x, 0, 30}];
```

```
f[x_] = solution[[1,1,2]];
graph[i] = Plot[f[x], {x,0,30}, PlotStyle ->Hue[.2 i + .5], DisplayFunction
   ->Identity]}, {i, -2, 2}]
Show[graph[-2], graph[-1], graph[0], graph[1], graph[2], DisplayFunction
   -> $DisplayFunction, PlotRange ->All];
```



实际中是用的彩色绘制这五条曲线，因此可以很容易地区分开它们.

11.3　拉普拉斯变换

本节我们讲述一种求解微分方程的巧妙的方法. 虽然这里讲解的过程适用于更广泛的一类问题，但它真正的能力就在于能求解更一般的微分方程，即使方程右边为不连续情形，或者右边除了在很小区间上值很大外，其余点都为零. 由于绝大多数的这类问题来自于以时间为独立变量的场景，因此通常就把 y 以及其导数表示成 t 的函数. 我们下面只是启发性地给出拉普拉斯变换满足的性质，而不去讨论这些性质成立的条件.

如果 f 为定义在区间 $[0,\ \infty)$ 上的函数，$f(t)$ 的拉普拉斯变换定义为

$$\mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t)\,dt,$$

其之所以非常有用，就在于有下述性质成立(不加证明)：

$\mathcal{L}\{1\} = \dfrac{1}{s}$　　　$\mathcal{L}\{\sinh(bt)\} = \dfrac{b}{s^2 - b^2}$

$\mathcal{L}\{t\} = \dfrac{1}{s^2}$　　　$\mathcal{L}\{\cosh(bt)\} = \dfrac{b}{s^2 - b^2}$

$\mathcal{L}\{t^n\} = \dfrac{n!}{s^{n+1}}$，$n$ 为正整数　　　$\mathcal{L}\{e^{at}\sinh(bt)\} = \dfrac{b}{(s-a)^2 - b^2}$

$\mathcal{L}\{e^{at}\} = \dfrac{1}{s-a}$　　　$\mathcal{L}\{e^{at}\cosh(bt)\} = \dfrac{s-a}{(s-a)^2 - b^2}$

$\mathcal{L}\{\sin(bt)\} = \dfrac{b}{s^2 + b^2}$　　　$\mathcal{L}\{f'(t)\} = s\,\mathcal{L}\{f(t)\} - f(0)$

$\mathcal{L}\{\cos(bt)\} = \dfrac{s}{s^2 + b^2}$　　　$\mathcal{L}\{f''(t)\} = s^2\,\mathcal{L}\{f(t)\} - sf(0) - f'(0)$

$\mathcal{L}\{e^{at}\sin(bt)\} = \dfrac{b}{(s-a)^2 + b^2}$　　　$\mathcal{L}\{af(t) + bg(t)\} = a\,\mathcal{L}\{f(t)\} + b\,\mathcal{L}\{g(t)\}$；

$\mathcal{L}\{e^{at}\cos(bt)\} = \dfrac{s-a}{(s-a)^2 + b^2}$　　　如果 $F(s) = \mathcal{L}\{f(t)\}$，那么 $\mathcal{L}\{e^{at}f(t)\} = F(s-a)$

在 Mathematica 中调用 **LaplaceTransform** 命令就可以计算函数 f 的拉普拉斯变换.

■ **LaplaceTransform[f[变量1], 变量1, 变量2]**计算独立变量为变量1的函数 f 的拉普拉斯变换，并把结果表示为变量2的函数.

> **注意** 在 Mathematica 版本 4 中，LaplaceTransform 以及本节后面要讲到的 Inverse-LaplaceTransform、UnitStep 与 DiracDelta 命令都包含在内核中，而在版本 3 中，这些命令包含在软件包 Calculus`LaplaceTransform`与 Calculus`DiracDelta`中. 如果读者用的是版本 3，那么在使用这些命令之前必须上载相应软件包.

例 17 下述命令的结果与上面所列的性质吻合.

LaplaceTransform[Exp[2t] Sin[3t], t, s]

$\frac{3}{13-4s+s^2}$

LaplaceTransform[Exp[2t] Cos[3t], t, s]

$\frac{-2+s}{13-4s+s^2}$

LaplaceTransform[a f[t] + b g[t], t, s]

a LaplaceTransform[f[t], t, s] + b LaplaceTransform[g[t], t, s]

LaplaceTransform[f'[t], t, s]

- f[0] + s LaplaceTransform[f[t], t, s]

LaplaceTransform[f''[t], t, s]

$-$ s f[0] + s^2 LaplaceTransform[f[t], t, s] $-$ f'[0]

拉普拉斯变换的能力来源于它建立了 $f(t)$ 与 $\mathcal{L}\{f(t)\}$ 之间的一一对应. 这就是说如果知道了 $\mathcal{L}\{f(t)\}$，那么 $f(t)$ 是惟一确定的. 如果 $F(s)=\mathcal{L}\{f(t)\}$，那么 $f(t)=\mathcal{L}^{-1}\{F(s)\}$，$\mathcal{L}^{-1}$ 称为反拉普拉斯变换.

- **InverseLaplaceTransform[F[变量 1], 变量 1, 变量 2]** 计算独立变量为变量 1 的函数 F 的反拉普拉斯变换，把结果表示为变量 2 的函数.

例 18 **InverseLaplaceTransform[$\frac{1}{s-3}$, s, t]**

e^{3t}

InverseLaplaceTransform[$\frac{1}{s^3-8}$, s, t]

$\frac{1}{12}e^{-t}(e^{3t}-\text{Cos}[\sqrt{3}t]-\sqrt{3}\text{Sin}[\sqrt{3}t])$

> 如果手工计算反变换，那么要对分母进行因式分解，然后把分式转化为部分分式，并计算其中每一项的反变换. 而 Mathematica 会自动进行这些运算.

下面这个例子说明了拉普拉斯变换是如何应用于简单微分方程求解的.

例 19 求解微分方程 $\frac{d^2y}{dt^2}-3\frac{dy}{dt}+2y=t^2, y'(0)=1, y(0)=2$.

首先计算微分方程两边的拉普拉斯变换. 这可以在一步内就完成.

equation = y''[t] - 3y'[t] + 2y[t] == t^2;

temp = LaplaceTransform[equation, t, s]

2 LaplaceTransform[y[t], t, s] + S^2 LaplaceTransform[y[t], t, s] $-$

3(s LaplaceTransform[y[t], t, s] $-$ y[0]) $-$ s y[0] $-$ Y'[0] == $\frac{2}{s^3}$

然后，求解在满足初值条件时拉普拉斯变换的表达式.

temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/. {y'[0]→1, y[0]→2}

$\left\{\left\{\text{LaplaceTransform}[y[t],t,s] \to -\dfrac{5-\frac{2}{s^3}-2s}{2-3s+s^2}\right\}\right\}$

现在从中提取 s 的函数.

```
temp3 = temp2[[1,1,2]]
```

$\dfrac{5-\frac{2}{s^3}-2s}{2-3s+s^2}$

最后,计算反拉普拉斯变换,以得到方程的解.

```
InverseLaplaceTransform[temp3, s, t]
```

$\frac{1}{4}(7+4e^{t}-3e^{2t}+6t+2t^2)$

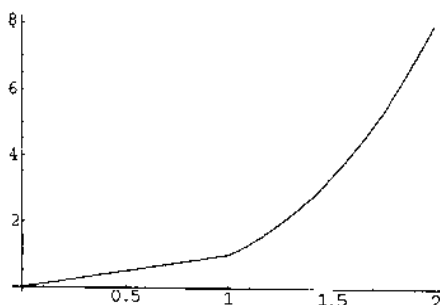
正如在本节开始指出的那样,拉普拉斯变换是求解包含不连续右边项的微分方程的理想工具.现在我们引入 Heaviside 函数,也称为阶梯函数,它对于后面的讲解非常有用.

■ **UintStep[x]** 如果 x<0,给出值 0,如果 x≥0,给出值 1.

我们在这里用 $u(t)$表示阶梯函数,利用这个函数表示分段定义的函数相当方便.

例 20　画出函数 $g(x)=\begin{cases}x, & \text{如果 } x<1,\\ x^3, & \text{如果 } x\geq 1\end{cases}$ 在 $0\leq x\leq 2$ 上的图形.

```
g[x_] = UnitStep[1 - x]x + UnitStep[x - 1]x^3;
Plot[g[x], {x, 0, 2}];
```



容易证明$\mathcal{L}\{u(t-c)\}=\dfrac{e^{-cs}}{s}$,而且如果 $F(s)=\mathcal{L}\{f(t)\}$,那么$\mathcal{L}\{u(t-c)f(t-c)\}=e^{-cs}F(s)$.这些性质使得可以很方便地求解包含分片定义函数的微分方程.

例21　求解微分方程 $\dfrac{d^2y}{dt^2}-3\dfrac{dy}{dt}+2y=g(t)$, $y(0)=y'(0)=0$, 其中 $g(t)=\begin{cases}1, & \text{如果 } 0\leq t\leq 1,\\ 0, & \text{如果 } t>1,\end{cases}$ 并画出解在 $0\leq t\leq 2$ 时的图形.

由于 $t\geq 0$,所以 $g(t)=$UnitStep[1 - t].

```
temp = LaplaceTransform[y''[t] - 3y'[t] + 2y[t] == UnitStep[1 - t], t, s]
```

$2\,\text{LaplaceTransform}[y[t],t,s]+s^2\,\text{LaplaceTransform}[y[t],t,s]-3\,(s\,\text{LaplaceTransform}[y[t],t,s]-y[0])-s\,y[0]-y'[0]==\dfrac{1-e^{-s}}{s}$

```
temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/.{y'[0]→0, y[0]→0}
```

$\{\{\text{LaplaceTransform}[y[t],t,s]\to\frac{1-e^{-s}}{s(2-3s+s^2)}\}\}$

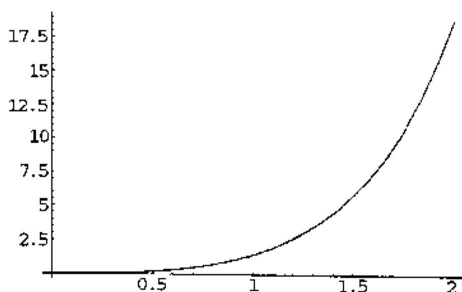
temp3 = temp2[[1,1,2]]

$\frac{1-e^{-s}}{s(2-3s+s^2)}$

f[t_] = InverseLaplaceTransform[temp3, s, t]

$\frac{-(e-e^t)^2\,\text{UnitStep}[-1+t]+e^2(-1+e^t)^2\,\text{UnitStep}[t]}{2e^2}$

Plot[f[t], {t, 0, 2}]



从解的图形中可见，即使方程中包含不连续的函数，解也可以是连续函数.

在物理学与生物学中，经常遇到一类微分方程，其右边项 $f(t)$ 是一个脉冲函数，即 $f(t)$ 除了在很小的时间区间内取正值外，其它点都取零值.

Dirac 的 δ 函数就是一个理想化的脉冲函数. 虽然从传统意义上看，这并不是一个真正的函数，但它的合理性是建立在 20 世纪中叶 Laurent Schwartz 创立的广义函数理论上的. 这个函数是由下述两个条件定义的：

$$\delta(t-t_0)=0, \text{如果 } t\neq t_0;$$

$$\int_{-\infty}^{\infty}\delta(t-t_0)\mathrm{d}t=1.$$

由定义很容易得到结论 $\int_{-\infty}^{\infty}f(t)\delta(t-t_0)\mathrm{d}t=f(t_0)$. 由此可知当 $t_0\geqslant 0$ 时，$\mathcal{L}\{\delta(t-t_0)\}=\int_0^{\infty}e^{-st}\delta(t-t_0)\mathrm{d}t=e^{-st_0}$，否则它的值就为零.

■ **DiracDelta[t]** 给出满足 $t\neq 0$，并且 $\int_{-\infty}^{\infty}\delta(t)\mathrm{d}t=1$ 时 $\delta(t)=0$ 的 Dirac δ 函数 $\delta(t)$.

例 22 $\int_{-\infty}^{\infty}$ **DiracDelta[t] dt**

1

$\int_{-\infty}^{\infty}$ **f[t]DiracDelta[t − a] dt**

f[a]

例 23 **LaplaceTransform[DiracDelta[t − a],t,s]**

e^{-as} UnitStep[a]

LaplaceTransform[DiracDelta[t − 3],t,s] ←由于 Mathematica 不知道 a 是负数还是正数，因此在拉普拉斯变换的结果中包含 UnitStep[a].

e^{-3s}

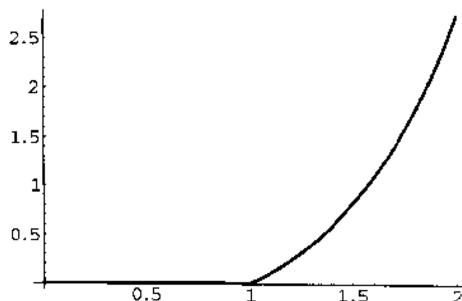
LaplaceTransform[DiracDelta[t + 3],t,s]

0

由于已经知道了 Dirac δ 函数的拉普拉斯变换,因此我们可以按照与例 19 中描述的方法几乎完全相同的过程求解包含脉冲函数的微分方程. 下面这个例子演示了这种方法.

例 24 求出微分方程$\frac{d^2y}{dt^2}-2\frac{dy}{dt}+y=\delta(t-1)$, $y(0)=y'(0)=0$的解.

```
equation = y''[t] - 2y'[t] + y[t] == DiracDelta[t - 1];
temp = LaplaceTransform[equation, t, s]
LaplaceTransform[y[t], t, s] + s^2 LaplaceTransform[y[t], t, s] -
  2 (s LaplaceTransform[y[t], t, s] - y[0]) - sy[0] - y'[0] == e^-s
temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/.{y'[0]->0, y[0]->0}
{{LaplaceTransform[y[t], t, s] -> e^-s/(-1 + s)^2}}
temp3 = temp2[[1, 1, 2]]//Expand
e^-s/(-1 + s)^2
solution[t_] = InverseLaplaceTransform[temp, s, t]
e^(-1+t) (-1 + t) UnitStep[-1 + t]
Plot[solution[t], {t, 0, 2}, PlotStyle -> Thickness[.01]];
```



拉普拉斯变换也可以用来求解微分方程组,方法与处理单个微分方程时类似,只是这时要对每个独立变量定义不同的变换. 下面的例子演示了求解由两个一阶微分方程构成的方程组的步骤. 自然可以推广到更多的方程组以及更高阶的情形.

例 25 求解微分方程组$\begin{cases}\frac{dx}{dt}+y=t\\ 4x+\frac{dy}{dt}=0\end{cases}$,初值条件为 $x(0)=1$, $y(0)=-1$.

```
system = {x'[t] + y[t] == t, 4x[t] + y'[t] == 0};
temp = LaplaceTransform[system, t, s]
s LaplaceTransform[x[t], t, s] + LaplaceTransform[y[t], t, s] - x[0] == 1/s^2, 4
     LaplaceTransform[x[t], t, s] + s LaplaceTransform[y[t], t, s] - y[0] == 0
temp2 = Solve[temp, {LaplaceTransform[x[t], t, s],
      LaplaceTransform[y[t], t, s]}]/.{x[0]->1, y[0]-> -1}
{{LaplaceTransform[x[t], t, s] -> -(-1 - s - s^2)/(s(-4 + s^2)),
  LaplaceTransform[y[t], t, s] -> -(4 + 4s^2 + s^3)/(s^2(-4 + s^2))}}
temp3a = temp2[[1, 1, 2]]
```

$-\frac{-1-s-s^2}{s(-4+s^2)}$

```
temp3b = temp2[[1,2,2]]
```

$-\frac{4+4s^2+s^3}{s^2(-4+s^2)}$

```
InverseLaplaceTransform[temp3a, s, t]
```

$\frac{1}{8}(-2+3\,e^{-2t}+7\,e^{2t})$

```
InverseLaplaceTransform[temp3b, s, t]
```

$\frac{3\,e^{-2t}}{4}-\frac{7\,e^{2t}}{4}+t$

所以方程组的解为 $x=\frac{1}{8}(-2+3e^{-2t}+7e^{2t})$, $y=\frac{3e^{-2t}}{4}-\frac{7e^{2t}}{4}+t$.

习题解答

11.18 求解微分方程$\frac{d^2y}{dt^2}+y=\sin t$，初值条件为 $y(0)=0$, $y'(0)=2$.

解 ☞

```
equation = y''[t] + y[t] == Sin[t];
temp = LaplaceTransform[equation, t, s]
```

$\text{LaplaceTransform}[y[t],t,s]+s^2\,\text{LaplaceTransform}[y[t],t,s]-s\,y[0]-y'[0]==\frac{1}{1+s^2}$

```
temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/.{y'[0]→2,y[0]→0}
```

$\left\{\left\{\text{LaplaceTransform}[y[t],t,s]\to-\frac{-2-\frac{1}{1+s^2}}{1+s^2}\right\}\right\}$

```
temp3 = temp2[[1,1,2]];
InverseLaplaceTransform[temp3, s, t]
```

$\frac{1}{2}(-t\,\text{Cos}[t]+5\,\text{Sin}[t])$

11.19 求解微分方程$\frac{d^2y}{dt^2}+\frac{dy}{dt}+y=e^t$, $y(0)=3$, $y'(0)=2$.

解 ☞

```
equation = y''[t] + y'[t] + y[t] == Exp[t];
temp = LaplaceTransform[equation, t, s]
```

$\text{LaplaceTransform}[y[t],t,s]+s\,\text{LaplaceTransform}[y[t],t,s]+s^2\,\text{LaplaceTransform}[y[t],t,s]-y[0]-s\,y[0]-y'[0]==\frac{1}{-1+s}$

```
temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/.{y'[0]→2,y[0]→3}
```

$\left\{\left\{\text{LaplaceTransform}[y[t],t,s]\to-\frac{-5-\frac{1}{-1+s}-3s}{1+s+s^2}\right\}\right\}$

```
temp3 = temp2[[1,1,2]];
InverseLaplaceTransform[temp3, s, t]
```

$$\frac{1}{3}e^{-t/2}\left(e^{3t/2}+8\text{Cos}\left[\frac{\sqrt{3}t}{2}\right]+6\sqrt{3}\text{Sin}\left[\frac{\sqrt{3}t}{2}\right]\right)$$

11.20　求解微分方程 $\frac{d^2y}{dt^2}-2\frac{dy}{dt}+y=g(t)$, $y(0)=y'(0)=0$, 其中 $g(t)=\begin{cases}t, & \text{如果 } 0\leqslant t\leqslant 1,\\ t^2, & \text{如果 } t>1,\end{cases}$ 然后画出解在 $0\leqslant t\leqslant 4$ 时的图形.

解 ☞

```
equation = y''[t] - 2y'[t] + y[t] == t UnitStep[1 - t] + t^2 UnitStep[t - 1];
temp = LaplaceTransform[equation, t, s]
```

LaplaceTransform [y[t], t, s] + s² LaplaceTransform [y[t], t, s] − 2(s LaplaceTransform[y[t], t, s] − y[0]) − s y[0] − y'[0] ==

$$\frac{e^{-s}(-1+e^s-s)}{s^2}+\frac{e^{-s}(2+2s+s^2)}{s^3}$$

```
temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/.{y'[0]→0, y[0]→0}
```

$$\left\{\left\{\text{LaplaceTransform}[y[t],t,s]\to\frac{e^{-s}(2+s+e^s s)}{(-1+s)^2 s^3}\right\}\right\}$$

如果不展开拉普拉斯变换的结果, Mathematica 在计算反变换时会遇到麻烦.

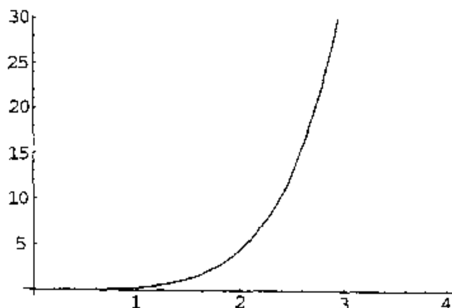
```
temp3 = temp2[[1,1,2]]//Expand
```

$$\frac{2e^{-s}}{(-1+s)^2s^3}+\frac{1}{(-1+s)^2s^2}+\frac{e^{-s}}{(-1+s)^2s^2}$$

```
f[t_] = InverseLaplaceTransform[temp3, s, t]
```

$$2+e^t(-2+t)+t+2\left(\frac{1}{2}(6+4(-1+t)+(-1+t)^2)+e^{-1+t}(-4+t)\right)\text{UnitStep}[-1+t]+(1+e^{-1+t}(-3+t)+t)\text{UnitStep}[-1+t]$$

```
Plot[f[t], {t, 0, 4}];
```



11.21　求解微分方程组 $\begin{cases}\frac{dx}{dt}+y=t\sin t,\\ x+\frac{dy}{dt}=t\cos t,\end{cases}$ $x(0)=y(0)=0$.

解 ☞

```
system = {x'[t] + y[t] == t Sin[t], x[t] + y'[t] == t Cos[t]};
temp = LaplaceTransform[sytem, t, s]
```

{sLaplaceTransform[x[t], t, s] + LaplaceTransform[y[t], t, s] − x[0] ==

$\frac{2s}{(1+s^2)^2}$, LaplaceTransform[x[t], t, s] +

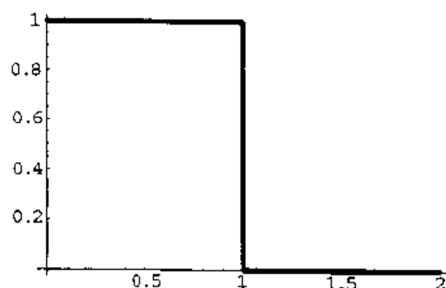
```
s LaplaceTransform[y[t], t, s] - y[0] == (-1 + s^2)/(1 + s^2)^2}
temp2 = Solve[temp, {LaplaceTransform[x[t], t, s],
      LaplaceTransform[y[t], t, s]}]/.{x[0]→0, y[0]→0}
{{LaplaceTransform[x[t], t, s]→ 1/((-1 + s^2)(1 + s^2)),
  LaplaceTransform[y[t], t, s]→ -(2s/(1 + s^2)^2 - s(-1 + s^2)/(1 + s^2)^2)/(-1 + s^2)}}
temp3a = temp2[[1,1,2]];
temp3b = temp2[[1,2,2]];
InverseLaplaceTransform[temp3a, s, t]
1/4(-e^-t + e^t - 2 Sin[t])
InverseLaplaceTransform[temp3b, s, t]
1/4(-e^-t - e^t + 2Cos[t] + 4t Sin[t])
```

所以微分方程组的解为 $x=\frac{1}{4}(-e^{-t}+e^{t}-2\sin t)$, $y=\frac{1}{4}(-e^{-t}-e^{t}+2\cos t+4t\sin t)$.

11.22 在一个由电阻与电感组成的简单电路中,施加电动势 E(电压),那么电流 I 满足方程:$L\frac{dI}{dt}+RI=E$,其中 E,I 与 L 的单位分别为伏特、安培与亨利. 设 $R=10$ 欧姆, $L=1$ 亨利. 如果在时刻 $t=0$ 施加直流电压 1 伏特,1 秒后关闭,请画出在前 2 秒内电流的图形.

解 ☞

```
e[t_] = UnitStep[1 - t];
Plot[e[t], {t, 0, 2}, PlotStyle →Thickness[.01]];
```



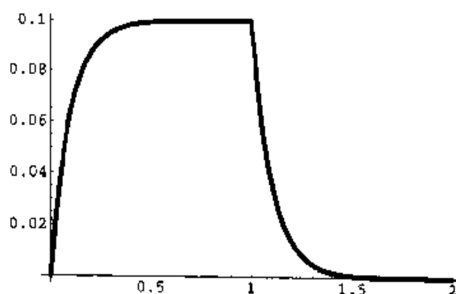

```
l = 1; r = 10;
equation = l i'[t] + r i[t] == e[t];
temp = LaplaceTransform[equation, t, s]
-i[0] + 10LaplaceTransform[i[t], t, s] + sLaplaceTransform[i[t], t, s] ==
(1 - e^-s)/s
temp2 = Solve[temp, LaplaceTransform[i[t], t, s]]/.i[0]→0
{{LaplaceTransform[i[t], t, s]→ e^-s(-1 + e^s)/(s(10 + s))}}
temp3 = temp2[[1,1,2]]//Expand
```

$$\frac{1}{s(10+s)}-\frac{e^{-s}}{s(10+s)}$$

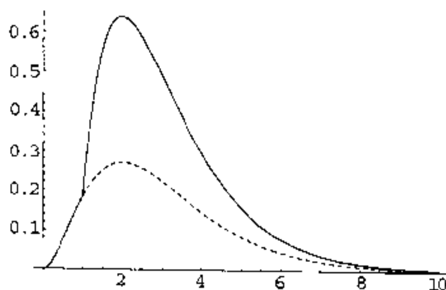
```
f[t_] = InverseLaplaceTransform[temp3, s, t]
```

$$\frac{1}{10}(1-e^{-10t})-\frac{1}{10}(1-e^{-10(-1+t)})\text{UnitStep}[-1+t]$$

```
Plot[f[t], {t, 0, 2}, PlotStyle → Thickness[.01]];
```



11.23　弹簧的一端有质量为 m 的质点,另一端被吊挂起来,这样它就会处于平衡位置不动.如果对质点施加外力 $f(t)$,那么质点的运动遵从方程 $m\frac{d^2y}{dx^2}+a\frac{dy}{dt}+ky=0$. $y'(0)=0$, $y(0)=0$,其中 a 是阻尼常数,k 为弹簧的弹性系数.假设 $m=1$, $a=2$, $k=1$, $f(t)=e^{-t}$,请描述弹簧的运动情形.然后再确定在一秒钟后加上1lb·sec冲量后的弹簧的运动情形.在同一坐标系中画出这两种运动的图形.



解 ☞

```
m = 1; a = 2; k = 1;
equation = my″[t] + ay′[t] + ky[t] == Exp[-t];←没有冲量.
temp = LaplaceTransform[equation, t, s],
temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/.{y′[0]→0, y[0]→0};
temp3 = temp2[[1,1,2]]//Expand;
Print["Solution Without Impulse (dashed)"]
f1[t_] = InverseLaplaceTransform[temp3, s, t]
g1 = Plot[f1[t], {t, 0, 10}, PlotStyle → Dashing[{.01}],
  DisplayFunction → Identity];
equation = m y″[t] + a y′[t] + k y[t] == Exp[-t] + DirecDelta[t - 1];←有冲量.
temp = LaplaceTransform[equation, t, s];
temp2 = Solve[temp, LaplaceTransform[y[t], t, s]]/.{y′[0]→0, y[0]→0};
temp3 = temp2[[1,1,2]]//Expand;
Print["Solution With Impulse (solid)"]
f2[t_] = InverseLaplaceTransform[temp3, s, t]
```

```
g2 = Plot[f2[t], {t, 0, 10}, DisplayFunction -> Identity];
```

Solution Without Impulse(dashed)

(没有冲量时的解(点划线))

$\frac{1}{2}e^{-t}t^2$

Solution With Impulse(solid)

(有冲量时的解(实线))

$\frac{1}{2}e^{-t}t^2 + e^{1-t}(-1+t)$ UnitStep[-1+t]

```
Show[g1, g2, DisplayFunction -> $DisplayFunction];
```

第十二章　线性代数

12.1　向量与矩阵

向量与矩阵在 Mathematica 中是用列表(见第三章中的介绍)表示的.所谓向量就是简单列表,而矩阵则是向量的列表.

向量或矩阵可以作为列表以手工方式输入其中的元素,也可以采用内置命令更方便地输入其中的元素.

向量

- **Table[表达式, {i, n}]** 构造一个 n 维向量,其中的元素由表达式在 $i = 1, 2, 3, \cdots, n$ 时的值组成.
- **Array[f, n]** 构造一个 n 维向量,其中的元素分别为 f[1],f[2],…,f[n],这里 f 为一元函数.

矩阵

- **Table[表达式, {i, m}, {j,n}]** 构造一个 $m \times n$ 阶矩阵,其中的元素由表达式在 $(i, j) = (1, 1), \cdots, (m, n)$ 时的值组成.
- **Array[f, {m, n}]** 构造一个 $m \times n$ 阶矩阵,其中的元素分别为 f[1,1],…,f[m,n],这里 f 为二元函数.
- **DiagonalMatrix[列表]**生成一个对角矩阵,对角线上的元素由给定列表(一维列表)中的元素构成.
- **IdentityMatrix[n]** 创建一个 $n \times n$ 阶单位阵.

虽然矩阵是作为列表而创建的,但是可以利用 **MatrixForm** 命令把其中的内容显示为矩阵形式,以方便查看.

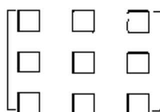
- MatrixForm[**列表**] 在矩形方阵中显示出列表中的元素.如果列表只是一维的,那么 MatrixForm 就把它显示为列向量(即作为 $n \times 1$ 阶矩阵).

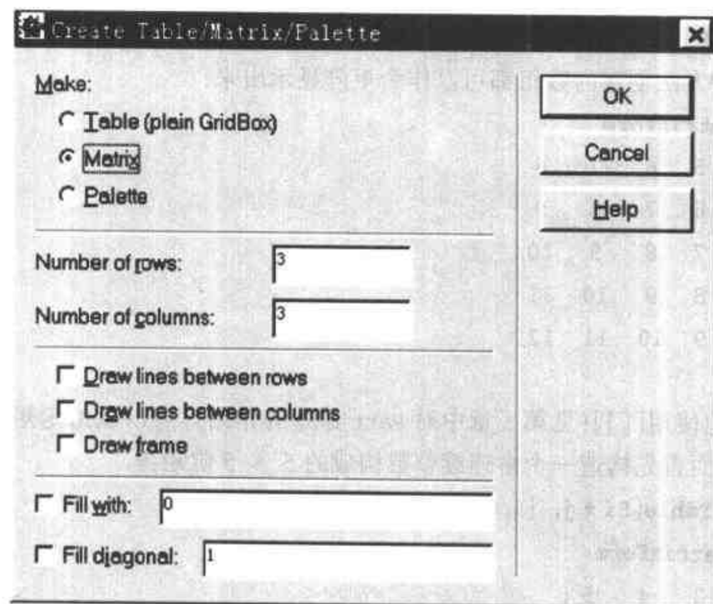
在列表的右边输入**//MatrixForm** 与 MatrixForm[列表]是等价的,但是相比起来,前者更方便一些.然而在使用前者时必须仔细一些,注意不要在矩阵的定义中使用//MatrixForm 命令.只有为了以矩阵形式显示列表中的元素时,才需要使用这条命令.

例 1
```
m = {{1, 1}, {1,2}};
m//MatrixForm
```
$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

另外,矩阵也可以通过菜单命令 **Input ⇒Create Table/Matrix/Palette**(创建表格、矩阵或模板)来输入其中的元素.

通过上述菜单命令,就会得到如右图所示的网格.一旦建立起这样的网格,就可以在其中方便地输入数字,并可以使用 **TAB** 键从一个空格移到另一个空格中.这个菜单命令提供了用 0 和 1 填充矩阵的选项,从而可以非常方便地创建稀疏矩阵.





例 2 在这个例子中创建一个向量，它的元素由前五个连续正整数的平方构成．为此不妨采用直接输入的方法：

```
squares = {1, 4, 9, 16, 25}
{1, 4, 9, 16, 25}
```

然而更简便的方法是利用 **Table** 命令或者 **Array** 命令．

```
squares =Table[i^2, {i, 5}]
{1, 4, 9, 16, 25}
f[i_] =i^2;
squares =Array[f, 5]
{1, 4, 9, 16, 25}
```

如果要查看这个向量的内容，可以输入

MatrixForm[squares] 或者 **squares //MatrixForm**

$$\begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \end{bmatrix}$$

例 3 下面构造一个 7 × 5 阶矩阵，其元素由该元素所在位置的行指标与列指标的和构成．例如，$a_{23}=5$．对这个矩阵，当然可以利用在菜单 **Input ⇒Create Table /Matrix /Palette** 中的工具直接输入它的元素，但是采用一个标准的 Mathematica 矩阵创建命令可能会更好．下面是创建这个矩阵的两种方法．

```
matrix = Table[i + j, {i, 5}, {j,7}]
{{2, 3, 4, 5, 6, 7, 8}, {3, 4, 5, 6, 7, 8, 9},
 {4, 5, 6, 7, 8, 9, 10}, {5, 6, 7, 8, 9, 10, 11},
 {6, 7, 8, 9, 10, 11, 12}}
```

```
f[i_, j_] = i + j;
matrix = Array[f, {5,7}]
{{2, 3, 4, 5, 6, 7, 8}, {3, 4, 5, 6, 7, 8, 9},
 {4, 5, 6, 7, 8, 9, 10}, {5, 6, 7, 8, 9, 10, 11},
 {6, 7, 8, 9, 10, 11, 12}}
```

对于这两种方法创建的数组都可以作为矩阵显示出来.

```
matrix//MatrixForm
```

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

例 4　如果仔细地使用[[]](见第三章中对 Part 命令的介绍),可以从给定矩阵中构造出一个子矩阵.下面首先构造一个由连续整数构成的 5 × 5 阶矩阵.

```
matrix = Table[5i + j, {i,0,4}, {j,1,5}];
matrix//MatrixForm
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

这时候可以调用这个矩阵中每个具体的元素,例如调用它们在第三行第四列的元素.

```
matrix[[3,4]]
14
```

如果要调用整个第四行的元素,那么可以从 matrix 中提取第四个子列表.

```
matrix[[4]]
{16, 17, 18, 19, 20}
```

而整个第四列的元素可以借助于 All 指令来得到.

```
matrix[[All, 4]]
{4, 9, 14, 19, 24}
```

我们也可以得到在第 1, 3, 5 行、第 2, 4 列中的元素.

```
matrix[[{1,3,5}, {2,4}]]//MatrixForm
```

$$\begin{bmatrix} 2 & 4 \\ 12 & 14 \\ 22 & 25 \end{bmatrix}$$

或者通过删除 matrix 的第 2,4 行,得到一个 3 × 5 阶矩阵.

```
matrix[[{1,3,5}, All]]//MatrixForm
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 11 & 12 & 13 & 14 & 15 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

通过仔细地使用 Take 命令(第三章),我们甚至可以构造由 matrix 的位于第 2 行到第 4 行,第 3 列到第 5 列的元素组成的子矩阵.

Take[matrix, {2,4}, {3,5}]//MatrixForm

$$\begin{bmatrix} 8 & 9 & 10 \\ 13 & 14 & 15 \\ 18 & 19 & 20 \end{bmatrix}$$

习 题 解 答

12.1 构造一个由 2 的幂次组成的 10 维向量.

解 ☞

powersof2 = Table[2^k, {k, 1, 10}]

{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024}

powersof2//MatrixForm

$$\begin{bmatrix} 2 \\ 4 \\ 8 \\ 16 \\ 32 \\ 64 \\ 128 \\ 256 \\ 512 \\ 1024 \end{bmatrix}$$

12.2 构造一个由随机数字组成的 5 × 5 阶矩阵.

解 ☞

Table[Random[Integer, {0, 9}], {i, 5}, {j, 5}]//MatrixForm

$$\begin{bmatrix} 5 & 7 & 9 & 9 & 4 \\ 2 & 8 & 6 & 9 & 7 \\ 1 & 0 & 8 & 2 & 2 \\ 7 & 8 & 8 & 8 & 1 \\ 9 & 2 & 2 & 3 & 4 \end{bmatrix}$$

12.3 如果在矩阵定义中包含了**//MatrixForm**命令,会出现什么情况呢?

解 ☞

m = {{1,1},{1,2}}//MatrixForm

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$

m + m

$$2\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$$ ←并没有得到矩阵对应元素的和.

由于 m 不是列表,因此 Mathematica 不能进行指定的操作.为了看得更清楚些,我们使用 FullForm 命令.

FullForm[m]

MatrixForm[List[List[1,1], List[1,2]]]

现在按正确的方法操作：

```
m = {{1,1},{1,2}}
```

{{1,1},{1,2}}

```
m + m//MatrixForm
```

$$\begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix}$$

12.4 构造一个 10 × 10 阶对角矩阵，其对角元素由前 10 个素数构成.

解 ☞

```
primelist = Array[Prime, 10];
DiagonalMatrix[primelist]//MatrixForm
```

> **Prime** 是 Mathematica 内置的命令.

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 11 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 13 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 17 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 19 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 23 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 29 \end{bmatrix}$$

12.5 构造一个 5 × 5 阶上三角矩阵，其上三角元素都是 1，对主角线下面的元素都是 0.

解 ☞

```
m = Table[If[i ≤ j,1,0], {i,5},{j,5}];
m//MatrixForm
```

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

12.6 构造一个 7 × 7 阶准对角矩阵，其主对角线上元素都是 2，与主对角线相邻的元素都是 1，其它元素都是 0.

解 ☞

```
m = Table[If[Abs[i - j] == 1, 1, If[i == j,2,0]], {i,1,7},{j,1,7}];
m//MatrixForm
```

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

12.7 令 m 为 6 × 6 阶矩阵,元素由 1 到 36 的整数构成.从这个矩阵构造子矩阵 **m2**,它由 **m** 的位于奇数行与偶数列的元素组成.

解 ☞

```
m = Table[6i + j, {i,0,5},{j,1,6}];
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 \end{bmatrix}$$

```
m2 = m[[{1,3,5}, {2,4,6}]];
m2//MatrixForm
```

$$\begin{bmatrix} 2 & 4 & 6 \\ 14 & 16 & 18 \\ 26 & 28 & 30 \end{bmatrix}$$

12.2 矩阵运算

由于在 Mathematica 中向量与矩阵是作为列表来存贮的,因此在第三章中描述的所有列表操作都可以应用到向量与矩阵上.另外,还有一些专门的命令,只适用于矩阵.由于 n 维向量可以认为是 $n \times 1$ 阶矩阵,所以这些讲述的命令也同样适用于向量.在下面的语法描述中,m、m1 与 m2 表示矩阵,v1 与 v2 表示向量.

- **m1 + m2** 计算矩阵的和.
- **m1 - m2** 计算矩阵的差.
- **c m** 把矩阵 m 中每个元素都乘上常量 c.
- **m1.m2** 计算 m1 与 m2 的矩阵乘积.**v1.v2** 计算 v1 与 v2 的内积.对于矩阵而言,这个运算返回一个列表,而对于向量,只返回一个数.
- **Cross[v1, v2]**返回 **v1** 与 **v2** 的叉乘.(这只适用于三维向量.)通过输入(不加空格)按键序列 [Esc] **cross** [Esc] 也可以在计算表达式中插入叉乘符号 ×.

例 5 首先以列表形式生成一个 3 × 3 阶的随机矩阵.

```
m1 = Table[Random[Integer, {0, 9}], {i,1,3}, {j,1,3}]
{{9,4,2}, {2,9,3}, {0,1,4}}
m2 = Table[Random[Integer, {0, 9}], {i,1,3}, {j,1,3}]
{{2,8,1}, {8,3,4}, {6,4,0}}
```

下面以矩阵形式查看其内容.

```
m1//MatrixForm
```

$$\begin{bmatrix} 9 & 4 & 2 \\ 2 & 9 & 3 \\ 0 & 1 & 4 \end{bmatrix}$$

```
m2//MatrixForm
```

$$\begin{bmatrix} 2 & 8 & 1 \\ 8 & 3 & 4 \\ 6 & 4 & 0 \end{bmatrix}$$

下面的操作是把 m1 中每个元素都乘以 5.

```
5 m1
```

$$\begin{bmatrix} 45 & 20 & 10 \\ 10 & 45 & 15 \\ 0 & 5 & 20 \end{bmatrix}$$

接下来计算两个矩阵的和、差与积.

m1 + m2 //MatrixForm

$$\begin{bmatrix} 11 & 12 & 3 \\ 10 & 12 & 7 \\ 6 & 5 & 4 \end{bmatrix}$$

m1 - m2 //MatrixForm

$$\begin{bmatrix} 7 & -4 & 1 \\ -6 & 6 & -1 \\ -6 & -3 & 4 \end{bmatrix}$$

m1.m2 //MatrixForm

$$\begin{bmatrix} 62 & 92 & 25 \\ 94 & 55 & 38 \\ 32 & 19 & 4 \end{bmatrix}$$

必须注意在计算矩阵乘积时不能用 * 符号,因为这个符号根据列表中的约定,指的是把矩阵对应元素相乘.

m1 * m2//MatrixForm

$$\begin{bmatrix} 18 & 32 & 2 \\ 16 & 27 & 12 \\ 0 & 4 & 0 \end{bmatrix}$$

例 6 **v1 = {1,2,3};**

v2 = {4,5,6};

v1.v2

32　　　←在 Mathematica 中是用一个数表示内积,而并不用包含一个数的列表.

Cross[v1,v2]

{-3, 6, -3}

Mathematica 对行向量与列向量不加区分.因此,如果 v 是 n 维向量,m 是一个 $n \times n$ 阶矩阵,那么 v.m 与 m.v 都是有定义的(虽然它们一般得到不同的结果).而且,如果 v1 是一个 $n \times 1$ 阶矩阵(列向量),v2 是一个 $1 \times n$ 阶矩阵(行向量),那么 v1.v2 应当为 $n \times n$ 阶矩阵.但 Mathematica 仍然作为内积来计算.如果要计算两个向量的外积,应当使用命令 **Outer**.

■ **Outer[Times, v1, v2]**计算 v1 与 v2 的外积.

例 7 **v1 = {1,2,3};**

v2 = {4,5,6};

m = {{1,2,2}, {2,3,3}, {3,1,2}};

m//MatrixForm

$$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 3 & 3 \\ 3 & 1 & 2 \end{bmatrix}$$

```
m.v1
{11, 17, 11}
```

$$\begin{bmatrix}1&2&2\\2&3&3\\3&1&2\end{bmatrix}\begin{bmatrix}1\\2\\3\end{bmatrix}=\begin{bmatrix}11\\17\\11\end{bmatrix}$$

```
v1.m
{14, 11, 14}
```

$$[1\ 2\ 3]\begin{bmatrix}1&2&2\\2&3&3\\3&1&2\end{bmatrix}=[14\ 11\ 14]$$

```
Outer[Times, v1, v2]//MatrixForm
```

$$\begin{bmatrix}1\\2\\3\end{bmatrix}[4\ 5\ 6]=\begin{bmatrix}4&5&6\\8&10&12\\12&15&18\end{bmatrix}$$

$$\begin{bmatrix}4&5&6\\8&10&12\\12&15&18\end{bmatrix}$$

- **Inverse[矩阵]** 计算矩阵的逆矩阵.
- **Det[矩阵]** 计算矩阵的行列式.
- **Transpose[矩阵]** 计算矩阵的转置.
- **Tr[矩阵]** 计算矩阵的迹.
- **MatrixPower[矩阵, n]** 计算矩阵的 n 次方.
- **Minors[矩阵]** 生成一个矩阵,其第 (i, j) 个元素就是从矩阵中删去第 $n-i+1$ 行与第 $n-j+1$ 列得到的子矩阵的行列式.
- **Minors[矩阵, k]**生成一个矩阵,其元素由矩阵所有可能的 $k \times k$ 阶子矩阵的行列式组成.(这时矩阵不必是方阵.)

例 8

$$\mathbf{m1} = \begin{bmatrix}1&2&2\\2&3&3\\3&4&5\end{bmatrix};$$

> 这个矩阵以及下面的那个矩阵都是用在 Input 菜单中的 Create Table/Matrix/Palette 命令生成的.

$$\mathbf{m2} = \begin{bmatrix}1&2&3&4\\5&6&7&8\\9&10&11&12\end{bmatrix};$$

```
Inverse[m1]//MatrixForm
```

$$\begin{bmatrix}-3&2&0\\1&1&-1\\1&-2&1\end{bmatrix}$$

```
Tr[m1]
9
MatrixPower[m1, 3]//MatrixForm
```

$$\begin{bmatrix}97&142&160\\151&221&249\\231&338&381\end{bmatrix}$$

```
Transpose[m2]//MatrixForm
```

$$\begin{bmatrix}1&5&9\\2&6&10\\3&7&11\\4&8&12\end{bmatrix}$$

```
Tr[m2]
```

18

在计算迹时,矩阵不必是方阵.

例 9
```
m = Table[a[i,j], {i,1,3}, {j,1,3}];
m//MatrixForm
```

$$\begin{pmatrix} a[1,1] & a[1,2] & a[1,3] \\ a[2,1] & a[2,2] & a[2,3] \\ a[3,1] & a[3,2] & a[3,3] \end{pmatrix}$$

```
Minors[m]//MatrixForm
```

$$\begin{pmatrix} -a[1,1]a[2,1]+a[1,1]a[2,2] & -a[1,3]a[2,1]+a[1,1]a[2,3] & -a[1,3]a[2,2]+a[1,2]a[2,3] \\ -a[1,2]a[2,1]+a[1,1]a[3,2] & -a[1,3]a[3,1]+a[1,1]a[3,3] & -a[1,3]a[3,2]+a[1,2]a[3,3] \\ -a[2,2]a[3,1]+a[2,1]a[3,2] & -a[2,3]a[3,1]+a[2,1]a[3,3] & -a[2,3]a[3,2]+a[2,2]a[3,3] \end{pmatrix}$$

习题解答

12.8 向量的(欧几里得)模就是其所有分量的平方和的平方根.由此计算向量 (1, 3, 5, 7, 9, 11, 13, 15) 的模.

解 ☞

```
v = Table[2k - 1, {k, 1, 8}]
{1, 3, 5, 7, 9, 11, 13, 15}
norm = √v.v
2 √170
```

12.9 证明在 $\mathbb{R}^3$ 中两个向量的叉乘垂直于构成它的两个向量.

解 ☞

令 $\boldsymbol{u}$ = (u1, u2, u3), $\boldsymbol{v}$ = (v1, v2, v3),首先计算出 $\boldsymbol{w} = \boldsymbol{u} \times \boldsymbol{v}$.然后验证 $\boldsymbol{w} \perp \boldsymbol{u}$ 与 $\boldsymbol{w} \perp \boldsymbol{v}$.这里要利用两个向量垂直(⊥)当且仅当它们的内积为 0 这一事实.

```
u = {u1, u2, u3};
v = {v1, v2, v3};
w = Cross[u, v]
{-u3 v2 + u2 v3, u3 v1 - u1 v3, -u2 v1 + u1 v2}
u.w//Expand
0
v.w//Expand
0
```

12.10 可以证明由 $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ 构成的平行六面体的体积等于 $|\boldsymbol{u}\cdot(\boldsymbol{v}\times\boldsymbol{w})|$.利用这个公式计算由 $i+2j-3k, 2i-5j+k, 3i+j+2k$ 形成的平行六面体的体积.($\boldsymbol{u}\cdot(\boldsymbol{v}\times\boldsymbol{w})$ 称为向量 $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ 的混合积.)

解 ☞

```
u = {1, 2, -3};
v = {2, -5, 1};
w = {3, 1, 2};
volume = Abs[u.Cross[v,w]]
```

```
64
```

12.11 令 $\boldsymbol{u}$ = (u1, u2, u3), $\boldsymbol{v}$ = (v1, v2, v3), $\boldsymbol{w}$ = (w1, w2, w3). 证明这三个向量的混合积为

$$\boldsymbol{u}\cdot(\boldsymbol{v}\times\boldsymbol{w})=\begin{vmatrix} u1 & u2 & u3 \\ v1 & v2 & v3 \\ w1 & w2 & w3 \end{vmatrix}.$$

解 ☞

```
u = {u1, u2, u3};
v = {v1, v2, v3};
w = {w1, w2, w3};
matrix = {u, v, w};
lhs = u.Cross[v,w]//Expand;
rhs = Det[matrix]//Expand;
lhs = = rhs
True
```

12.12 希耳伯特矩阵是一个方阵,其第 i 行第 j 列元素为 $\frac{1}{i+j-1}$. 请构造 6 阶的希耳伯特矩阵,并计算它的行列式与逆矩阵.

解 ☞

```
f[i_, j_] = 1/(i+j-1);
hilbert = Array[f, {6,6}];
hilbert//MatrixForm
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} \end{bmatrix}$$

```
Det[hilbert]
```

$$\frac{1}{186313420339200000}$$

```
Inverse[hilbert]//MatrixForm
```

$$\begin{bmatrix} 36 & -630 & 3360 & -7560 & 7560 & -2772 \\ -630 & 14700 & -88200 & 211680 & -220500 & 83160 \\ 3360 & -88200 & 564480 & -1411200 & 1512000 & -582120 \\ -7560 & 211680 & -1411200 & 3628800 & -3969000 & 1552320 \\ 7560 & -220500 & 1512000 & -3969000 & 4410000 & -1746360 \\ -2772 & 83160 & -582120 & 1552320 & -1746360 & 698544 \end{bmatrix}$$

12.13 构造一个表格,列出从 1 阶到 10 阶希耳伯特矩阵(见习题 12.12)的行列式.

解 ☞

```
Clear[hilbert]
f[i_, j_] = 1/(i + j - 1);
Do[hilbert[k] = Array[f, {k,k}], {k, 1, 10}];
TableForm[Table[{k, Det[hilbert[k]]//N}, {k, 1, 10}],
    TableHeadings →{None, {"k", "determinant"}}]
```

k	determinant
1	1.
2	0.0833333
3	0.000462963
4	1.65344×10^{-7}
5	3.7493×10^{-12}
6	5.3673×10^{-18}
7	4.8358×10^{-25}
8	2.73705×10^{-33}
9	9.72023×10^{-43}
10	2.16418×10^{-53}

由于行列式不是零,所以各阶希耳伯特矩阵都是可逆的. 实际上希耳伯特矩阵是一个经典的病态矩阵例子.

12.14 令 $M = \begin{bmatrix} \frac{1}{10} & \frac{2}{10} & \frac{7}{10} \\ \frac{3}{10} & \frac{3}{10} & \frac{4}{10} \\ \frac{5}{10} & \frac{4}{10} & \frac{1}{10} \end{bmatrix}$ (M 是一个随机矩阵),计算 $\lim\limits_{n\to\infty} M^n$.

解 ☞

$$m = \frac{1}{10}\begin{bmatrix} 1 & 2 & 7 \\ 3 & 3 & 4 \\ 5 & 4 & 1 \end{bmatrix};$$

```
Clear[n]
Limit[MatrixPower[m, n], n → ∞]//MatrixForm
```

$$\begin{bmatrix} \frac{47}{150} & \frac{23}{75} & \frac{19}{50} \\ \frac{47}{150} & \frac{23}{75} & \frac{19}{50} \\ \frac{47}{150} & \frac{23}{75} & \frac{19}{50} \end{bmatrix}$$

12.15 令 $A = \begin{bmatrix} 1 & 2 & -1 & -2 & 3 \\ 2 & 1 & 2 & -2 & 0 \\ 0 & 1 & -2 & 3 & -1 \\ 1 & -1 & 1 & 2 & -3 \\ -2 & -2 & 1 & 1 & 2 \end{bmatrix}$, $f(x) = x^5 + 2x^4 - x^3 + x^2 - 3x - 2$, 计算 $f(A)$.

解 ☞

$$a = \begin{bmatrix} 1 & 2 & -1 & -2 & 3 \\ 2 & 1 & 2 & -2 & 0 \\ 0 & 1 & -2 & 3 & -1 \\ 1 & -1 & 1 & 2 & -3 \\ -2 & -2 & 1 & 1 & 2 \end{bmatrix};$$

```
MatrixPower[a, 5] + 2 MatrixPower[a,4] - MatrixPower[a,3] +
  MatrixPower[a,2] - 3 a + 2 IdentityMatrix[5]//MatrixForm
```

$$\begin{bmatrix} -496 & -948 & -189 & 1776 & -1695 \\ -726 & -862 & 288 & 714 & -66 \\ -117 & 399 & -103 & -648 & 1233 \\ -174 & 324 & 315 & -1216 & 1875 \\ 1419 & 1068 & -267 & -702 & -1069 \end{bmatrix}$$

12.16 可以证明复数 $a+bi$ 与矩阵 $\begin{bmatrix} a & b \\ -b & a \end{bmatrix}$ 具有相同的代数性质.利用矩阵计算 $(2+3i)^5$,并利用复数算术验证结果的正确性.

解 ☞

$$a = \begin{bmatrix} 2 & 3 \\ -3 & 2 \end{bmatrix};$$

```
MatrixPower[a,5]//MatrixForm
```

$$\begin{bmatrix} 122 & -597 \\ 597 & 122 \end{bmatrix}$$ ←这表示复数 122 - 597i.

```
(2 + 3I)^5
122 - 597 i
```

12.17 计算行列式

$$\begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \cdots & x_n^{n-1} \end{vmatrix}$$

在 $n=2, 3, 4, 5$ 时的值.由此可以归纳出结果的形式吗?这个行列式称为范德蒙行列式.

解 ☞

```
m[n_] := Table[x[i]^j, {j, 0, n-1}, {i, 1, n}];
m[2]//MatrixForm
```

$$\begin{bmatrix} 1 & 1 \\ x[1] & x[2] \end{bmatrix}$$

```
m[3]//MatrixForm
```

$$\begin{bmatrix} 1 & 1 & 1 \\ x[1] & x[2] & x[3] \\ x[1]^2 & x[2]^2 & x[3]^3 \end{bmatrix}$$

```
Det[m[2]]//Factor
-x[1]+x[2]
Det[m[3]]//Factor
-(x[1] - x[2]) (x[1] - x[3]) (x[2] - x[3])
Det[m[4]]//Factor
(x[1] - x[2]) (x[1] - x[3]) (x[2] - x[3]) (x[1] - x[4])
(x[2] - x[4]) (x[3] - x[4])
Det[m[5]]//Factor
(x[1] - x[2]) (x[1] - x[3]) (x[2] - x[3]) (x[1] - x[4])
(x[2] - x[4]) (x[3] - x[4]) (x[1] - x[5]) (x[2] - x[5])
(x[3] - x[5]) (x[4] - x[5])
```

一般地, $\det[m[n]] = \prod_{i<j} (x[j]-x[i])$.

12.18 在线性代数中有一个定理,内容是矩阵的行列式等于在任一行或任一列中每个元素与其对应的代数余子式乘积的和.(a_{ij}的代数余子式C_{ij}是$(-1)^{i+j}M_{ij}$,其中M_{ij}为对应的余子式.)利用这个定理计算一个随机生成的5 × 5阶矩阵的行列式,并检验所得结果的正确性.

解 ☞

```
n = 5;
a = Table[Random[Integer, {0, 9}], {i, 1, n}, {j,1,n}];
a//MatrixForm
```

$$\begin{bmatrix} 4 & 6 & 5 & 3 & 3 \\ 5 & 3 & 0 & 5 & 6 \\ 1 & 6 & 9 & 7 & 7 \\ 7 & 9 & 7 & 1 & 2 \\ 0 & 9 & 4 & 5 & 6 \end{bmatrix}$$

```
sigins = Table[(-1)^(i+j), {i, 1, n}, {j, 1, n}];
matrixofminors = Minors[a];
matrixofminors//MatrixForm
```

$$\begin{bmatrix} 171 & 159 & -174 & -283 & -216 \\ -1350 & -1584 & -270 & -140 & -48 \\ 669 & 549 & 342 & 293 & -78 \\ -561 & -339 & 96 & -143 & -168 \\ -3231 & -3333 & -438 & -497 & 246 \end{bmatrix}$$

```
cofactors = matrixofminors * signs;
cofactors//MatrixForm
```

$$\begin{bmatrix} 171 & -159 & -174 & 283 & -216 \\ 1350 & -1584 & 270 & -140 & 48 \\ 669 & -549 & 342 & -293 & -78 \\ 561 & -339 & -96 & -143 & 168 \\ -3231 & 3333 & -438 & 497 & 246 \end{bmatrix}$$

i = 3; ←对第三行进行展开.

$$\text{determinant} = \sum_{i=1}^{n} a[[n-i+1, n-j+1]] * \text{cofactors}[[i,j]]$$

```
2082
Det[a]
2082
```

12.19　令 $x=\begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix}$，计算 $x^Tx=[1\ 2\ 3\ 4\ 5]\begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix}$ 与 $xx^T=\begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix}[1\ 2\ 3\ 4\ 5]$.

解 ☞

x^Tx 是向量 x 与自身的内积，而 xx^T 是一个 5×5 阶矩阵.

$$x=\begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix};$$

```
x.x
55
Outer[Times, x, x]//MatrixForm
```

$$\begin{bmatrix}1&2&3&4&5\\2&4&6&8&10\\3&6&9&12&15\\4&8&12&16&20\\5&10&15&20&25\end{bmatrix}$$

12.3　矩阵的操作

在 Mathematica 中提供了一个软件包，名称为 **LinearAlgebra`MatrixManipulation`**，其中包含了大量矩阵操作命令，当处理线性代数中的问题时，这些命令就非常有用．当然，与其它软件包一样，必须在使用其中命令之前上载这个软件包.

下面的命令对给定矩阵进行增大处理：

- **AppendColumns[矩阵 1, 矩阵 2, …]** 通过把列合并在一起，由矩阵 1，矩阵 2，…组成一个矩阵.
- **AppendRows[矩阵 1, 矩阵 2, …]** 通过把行合并在一起，由矩阵 1，矩阵 2，…组成一个矩阵.
- **BlockMatrix[矩阵块]** 以分块的形式构成一个新矩阵．这里的矩阵块为一组矩阵，列表的构造确定块的安排方式.

例 10　**<<LinearAlgebra`MatrixManipulation`**

$$m1=\begin{bmatrix}a&b&c\\d&e&f\\g&h&i\end{bmatrix};$$

$$m2=\begin{bmatrix}aa&bb&cc\\dd&ee&ff\\gg&hh&ii\end{bmatrix};$$

```
AppendColumns[m1, m2]//MatrixForm
```

$$\begin{bmatrix} a & b & c \\ d & e & f \\ h & i & j \\ aa & bb & cc \\ dd & ee & ff \\ gg & hh & ii \end{bmatrix}$$

m1 与 m2 的对应列组合在一起.

AppendRow[m1,m2]//MatrixForm

$$\begin{bmatrix} a & b & c & aa & bb & cc \\ d & e & f & dd & ee & ff \\ g & h & i & gg & hh & ii \end{bmatrix}$$

m1 与 m2 的对应行组合在一起.

BlockMatrix[{{m1,m2}, {m2,m1}}]//MatrixForm

$$\begin{bmatrix} a & b & c & aa & bb & cc \\ d & e & f & dd & ee & ff \\ g & h & i & gg & hh & ii \\ aa & bb & cc & a & b & c \\ dd & ee & ff & d & e & f \\ gg & hh & ii & g & b & i \end{bmatrix}$$

列表的排列表明顶部有两块,从左到右分别是 m1 与 m2.底部也有两块,从左到右分别是 m2 与 m1

下面的命令可以用来构造子矩阵.

- **TakeRows[矩阵, n]** 构造一个由矩阵的前 n 行元素组成的子矩阵.如果 n 为负数,就用后 n 行元素构成子矩阵.
- **TakeRows[矩阵, {m,n}]** 由矩阵的第 m 行到第 n 行的元素构成子矩阵.
- **TakeColumns[矩阵, n]** 构造一个由矩阵的前 n 列元素组成的子矩阵.如果 n 为负数,就用后 n 列元素构成子矩阵.
- **TakeColumns[矩阵, {m,n}]** 由矩阵的第 m 列到第 n 列的元素构成子矩阵.
- **TakeMatrix[矩阵, {i1,j1}, {i2,j2}]** 构造一个由矩阵的介于(i1,j1)与(i2,j2)元素之间的元素组成的子矩阵.
- **SubMatrix[矩阵, {i,j},{m,n}]**由矩阵的从(i,j)位开始,构造一个 m × n 阶子矩阵.

例 11　**<<LinearAlgebra`MatrixManipulation`**

$$\mathbf{m} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \end{bmatrix};$$

TakeRows[m, 3]//MatrixForm

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$$

TakeColumns[m, - 3]//MatrixForm

$$\begin{bmatrix} 3 & 4 & 5 \\ 8 & 9 & 10 \\ 13 & 14 & 15 \\ 18 & 19 & 20 \end{bmatrix}$$

TakeRows[m, {2,3}]//MatrixForm

$$\begin{bmatrix} 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{bmatrix}$$

```
TakeColumns[m, {2,3}]//MatrixForm
```

$$\begin{bmatrix} 2 & 3 \\ 7 & 8 \\ 12 & 13 \\ 17 & 18 \end{bmatrix}$$

```
TakeMatrix[m,{2,2}, {3,4}]//MatrixForm
```

$$\begin{bmatrix} 7 & 8 & 9 \\ 12 & 13 & 14 \end{bmatrix}$$

```
SubMatrix[m, {2,3}, {2,2}]//MatrixForm
```

$$\begin{bmatrix} 8 & 9 \\ 13 & 14 \end{bmatrix}$$

虽然许多矩阵都可以用前一节介绍的 Table 或 Array 命令来创建,但是矩阵操作软件包中还是提供了一些命令,用来构造特定类型的矩阵.

- **UpperDiagonalMatrix[f,n]** 创建一个 n × n 阶上三角矩阵,主对角线及主对角线以上的元素为 f[i,j],其余元素为零.
- **LowerDiagonalMatrix[f,n]** 创建一个 n × n 阶下三角矩阵,主对角线及主对角线以下的元素为 f[i,j],其余元素为零.
- **ZeroMatrix[n]** 创建一个 n × n 阶零矩阵;**ZeroMatrix[m,n]**创建一个 m × n 阶零矩阵.
- **HilbertMatrix[n]** 创建一个 n × n 阶希耳伯特矩阵;**HilbertMatrix[m,n]**创建一个 m × n 阶希耳伯特矩阵.
- **HankelMatrix[n]** 创建一个 Hankel 矩阵,其第一行(与第一列)元素为{1, 2, 3, …, n};**HankelMatrix[n, 列表]** 创建一个 Hankel 矩阵,其第一行(与第一列)由列表中元素组成.

例 12

```
<<LinearAlgebra`MatrixManipulation`
f[i_, j_] = i+j;
LowerDiagonalMatrix[f,5]//MatrixForm
```

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 0 & 4 & 5 & 6 & 7 \\ 0 & 0 & 6 & 7 & 8 \\ 0 & 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

```
UpperDiagonalMatrix[f,5]//MatrixForm
```

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 3 & 4 & 0 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 \\ 5 & 6 & 7 & 8 & 0 \\ 6 & 7 & 8 & 9 & 10 \end{bmatrix}$$

```
ZeroMatrix[3,5]//MatrixForm
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
HilbertMatrix[5]//MatrixForm
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

```
HankelMatrix[5,{a,b,c,d,e}]//MatrixForm
```

$$\begin{bmatrix} a & b & c & d & e \\ b & c & d & e & 0 \\ c & d & e & 0 & 0 \\ d & e & 0 & 0 & 0 \\ e & 0 & 0 & 0 & 0 \end{bmatrix}$$

习题解答

12.20　构造一个 10 × 10 阶上三角矩阵,其非零元素为随机数字.证明这个矩阵的行列式等于其主对角线上元素的乘积.

解 ☞

```
<<LinearAlgebra`MatrixManipulation`
f[i_, j_] := Random[Integer, {0, 9}]
m = UpperDiagonalMatrix[f,10];
m//MatrixForm
```

$$\begin{bmatrix} 3 & 1 & 6 & 7 & 0 & 6 & 4 & 8 & 9 & 1 \\ 0 & 1 & 7 & 5 & 1 & 9 & 9 & 0 & 5 & 8 \\ 0 & 0 & 5 & 6 & 9 & 5 & 3 & 3 & 3 & 5 \\ 0 & 0 & 0 & 1 & 5 & 7 & 2 & 5 & 3 & 4 \\ 0 & 0 & 0 & 0 & 6 & 8 & 4 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 & 2 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 5 & 5 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 6 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

```
Det[m] == ∏_{i=1}^{10} m[[i,i]]
True
```

12.21　构造一个 9 × 9 阶块对角矩阵,其中一块为 2 × 2 阶矩阵,元素全是 2,另一块 3 × 3 阶矩阵,元素全是 3,最后一块 4 × 4 阶矩阵,元素全是 4.(所谓块对角矩阵,就是一个方阵,其对角线上的矩阵块也是方阵,而其它元素都是零.)

解 ☞

```
<<LinearAlgebra`MatrixManipulation`
```

```
m1 = Table[2, {2}, {2}];
m2 = Table[3, {3}, {3}];
m3 = Table[4, {4}, {4}];
BlockMatrix[{{m1, ZeroMatrix[2,7]},
             {ZeroMatrix[3,2], m2, ZeroMatrix[3,4]},
             {ZeroMatrix[4,5], m3}}]//MatrixForm
```

$$\begin{bmatrix} 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 \end{bmatrix}$$

12.22 令 $M = \begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \\ u & v & w & x & y \end{bmatrix}$,给出从 M 中删除第三行与第四行后得到的矩阵 P.

解 ☞

```
<<LinearAlgebra`MatrixManipulation`
temp = CharacterRange["a", "y"];        ←生成字母表.
m = Partition[temp, 5];       ←构造五个子列表,每个子列表中有五个字母.
m//MatrixForm
```

$$\begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \\ u & v & w & x & y \end{bmatrix}$$

```
temp1 = TakeColumns[m, {1,2}];          ← m的第1-2列.
temp2 = TakeColumns[m, {4,5}];          ←m的第4-5列.
temp3 = AppendRows[temp1, temp2];       ←把第1-2列与第4-5列组合起来.
temp3//MatrixForm
```

$$\begin{bmatrix} a & b & d & e \\ f & g & i & j \\ k & l & n & o \\ p & q & s & t \\ u & v & x & y \end{bmatrix}$$ ←这时矩阵的第3列已被删除,下面再删除其第4行.

```
temp4 = TakeRows[temp3, {1,3}];         ← temp3 的第1-2行.
temp5 = TakeRows[temp3, -1];            ←temp3 的第5行.
p = AppendRows[temp4, temp5];           ←把第1-2行与第5行组合起来.
p//MatrixForm
```

$$\begin{bmatrix} a & b & d & e \\ f & g & i & j \\ k & l & n & o \\ u & v & x & y \end{bmatrix}$$

12.4 线性方程组

Mathematica 提供了许多方法求解线性方程组. 在第六章中讨论的 Solve 命令就是其中的一种方法,但对于大的方程组而言,这种方法就显得有点儿麻烦,而且效率不高. 本节我们讨论用以求解线性方程组的其它的许多方法.

■ **LinearSolve[a, b]** 给出向量 **x**,使得 **a.x = b** 成立.

其中 a 是未知的系数矩阵,而 b 为线性方程组的"右边项". 如果 a 为可逆阵,那么 LinearSolve 就给出线性方程组的惟一解. 如果 a 为奇异阵,那么方程组就可能没有解,也有可能存在无穷组解.

如果方程组具有惟一解, Mathematica 就会给出这个解. 如果方程组没有解, Mathematica 就会返回出错信息.

例 13 方程组 $2x + y + z = 7, x - 4y + 3z = 2, 3x + 2y + 2z = 13$ 具有惟一解,而方程组 $2x + y + z = 7, x - 4y + 3z = 2, 3x - 3y + 4z = 10$ 没有解.

a1 = $\begin{bmatrix} 2 & 1 & 1 \\ 1 & -4 & 3 \\ 3 & 2 & 2 \end{bmatrix}$;

a2 = $\begin{bmatrix} 2 & 1 & 1 \\ 1 & -4 & 3 \\ 3 & -3 & 4 \end{bmatrix}$;

b1 = $\begin{bmatrix} 7 \\ 2 \\ 13 \end{bmatrix}$;

b2 = $\begin{bmatrix} 7 \\ 2 \\ 10 \end{bmatrix}$

```
LinearSolve[a1, b1]
{{1}, {2}, {3}}
LinearSolve[a2, b2]
LinearSolve::nosol :
Linear equation encountered which has no solution.
```

(遇到了没有解的线性方程组.)

```
LinearSolve[{{2,1,1}, {1,-4,3}, {3,-3,4}}, {{7}, {2}, {10}}]
```

如果方程组 a.x = b 有无穷组解,处理起来就有些复杂. 对于这种情况, Mathematica 就会返回一个解,称为特解. 而所有解则是由特解加上对应的齐次方程组 a.x = 0 的所有解构成的.

所有满足 a.x = 0 的向量 x 全体称为 a 的零空间,可以很容易地用 **NullSpace** 命令确定出来零空间的构造.

■ **NullSpace[a]** 返回 a 的零空间的基向量.

a 的零度, 即在 a 的零空间中最大线性无关向量组的向量个数可以通过计算 **Length[NullSpace[a]]** 得到. a 的秩可以用 **n - Length[NullSpace[a]]** 算出, 其中 n 表示 a 的

列数.

例 14　方程组 $2x+y+z=7, x-4y+3z=2, 3x-3y+4z=9$ 有无穷组解.

$$a=\begin{bmatrix}2 & 1 & 1\\ 1 & -4 & 3\\ 3 & -3 & 4\end{bmatrix};$$

$$b=\begin{bmatrix}7\\ 2\\ 9\end{bmatrix};$$

```
nullspacebasis = NullSpace[a]
{{-7, 5, 9}}
```

由于零空间非空,因此没有惟一解.

```
particular = LinearSolve[a, b]
```

$\left\{\left\{\frac{10}{3}\right\},\left\{\frac{1}{3}\right\},\{0\}\right\}$　　←这是一个特解.

方程组的完全解具有形式

```
t * nullspacebasis + particular
```

其中 t 是任意参数.然而,为了表示成单个列表,我们必须展平 nullspacebasis 和 particular.

```
generalsolution = t * Flatten[nullspacebasis] + Flatten[particular]
```

$\left\{\frac{10}{3}-7t, \frac{1}{3}+5t, 9t\right\}$

作为检验的步骤,我们下面把一般解代回到原来的方程组中.

```
a.x/.x→{10/3 - 7t, 1/3 + 5t, 9t}//Expand
{7, 2, 9}
```

求解线性方程组 **a.x = b** 的高斯-约当方法是基于对增广矩阵[**a** |**b**]的简化处理,这个过程通过初等行变换把矩阵转化为简化行梯形形式.

基本初等行变换有下述三种:

(a) 交换两行.

(b) 在一行中乘上一个非零常数.

(c) 在一行中加上另一行的倍数.

容易证明初等行变换对方程组的解没有影响.一个矩阵称为简化行梯形形式,是指

(a) 每一行中第一个非零元素都是 1(称为首 1).

(b) 在这个 1 的上面或下面的元素都是 0.

(c) 对于两个首 1 的行,下面的行中第一个非零元素比前一个的更靠右.

为了求解线性方程组,我们使用初等行变换把增广矩阵转化为简化行梯形形式.这时方程组有没有解,以及有解时解是什么就可以很容易得出.

学过线性代数的人都知道行变换是非常费时间的,而且整个过程非常繁烦,很容易出错.然而,Mathematica 的 **RowReduce** 命令可以自动地把任意矩阵转化为简化行梯形形式.

■　**RowReduce[矩阵]** 把矩阵转化为简化行梯形形式.

例 15　考虑 4 × 5 阶矩阵,其元素为 $a_{ij}=|i-j|$.

```
a = Table[Abs[i - j], {i, 1, 4}, {j, 1, 5}];
a //MatrixForm
```

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \end{bmatrix}$$

```
RowReduce[a]//MatrixForm
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{4}{3} \end{bmatrix}$$

下面演示行简化过程是如何用于线性方程组的求解的.为了便于比较,我们使用在前面的例 13 与例 14 中考虑的三个例子.

例 16 (a) 方程组 $2x + y + z = 7,\ x - 4y + 3z = 2,\ 3x + 2y + 2z = 13$ (具有惟一解)

(b) 方程组 $2x + y + z = 7,\ x - 4y + 3z = 2,\ 3x - 3y + 4z = 10$ (无解)

(c) 方程组 $2x + y + z = 7,\ x - 4y + 3z = 2,\ 3x - 3y + 4z = 9$ (有无穷组解)

这三个方程组的增广矩阵分别为

$$a1 = \begin{bmatrix} 2 & 1 & 1 & 7 \\ 1 & -4 & 3 & 2 \\ 3 & 2 & 2 & 13 \end{bmatrix};\quad a2 = \begin{bmatrix} 2 & 1 & 1 & 7 \\ 1 & -4 & 3 & 2 \\ 3 & -3 & 4 & 10 \end{bmatrix};\quad a3 = \begin{bmatrix} 2 & 1 & 1 & 7 \\ 1 & -4 & 3 & 2 \\ 3 & -3 & 4 & 9 \end{bmatrix};$$

```
RowReduce[a1]//MatrixForm
```

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

当矩阵被简化后,如果把它看作方程组,可知 $x = 1, y = 2, z = 3$.

```
RowReduce[a2]//MatrixForm
```

$$\begin{bmatrix} 1 & 0 & \frac{7}{9} & 0 \\ 0 & 1 & -\frac{5}{9} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

简化后的矩阵底行指的是 $0x + 0y + 0z = 1$,这当然是不可能的。这个矛盾(一行中只有最后一个是 1,其余都是 0)说明方程组没有解.

```
RowReduce[a3]//MatrixForm
```

$$\begin{bmatrix} 1 & 0 & \frac{7}{9} & \frac{10}{3} \\ 0 & 1 & -\frac{5}{9} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

底行全是 0 没有任何不对头的地方.然而这个方程组没有惟一解.如果令 $z = t$,其中 t 为独立参数,那么方程组的解具有如下形式:$x = \frac{10}{3} - \frac{7}{9}t,\ y = \frac{1}{3} + \frac{5}{9}t,\ z = t$.

注意 虽然这个解与例 14 给出的解在形式上稍有些不同,但从它们确定相同的解集的意义上看,它们是等价的.

另外一种相当常用的求解线性方程组的方法是 LU 分解法,特别当具有许多方程组,每个方程组的系数矩阵相同时,这种方法尤其有用.这个方法的基本思想是很简单的.

如果 A 为方阵,那么它可以分解为 $A = LU$,其中 L 为下三角矩阵,主对角线上元素全是 1,U 为上三角矩阵.这样方程组 $A\boldsymbol{x} = \boldsymbol{b}$ 就转化为 $(LU)\boldsymbol{x} = \boldsymbol{b}$,其可重写为$L(U\boldsymbol{x}) = \boldsymbol{b}$.如果令 $\boldsymbol{y} = U\boldsymbol{x}$,那么可以从 $L\boldsymbol{y} = \boldsymbol{b}$ 中解出 $\boldsymbol{y}$.一旦有了 $\boldsymbol{y}$,就可以从 $U\boldsymbol{x} = \boldsymbol{y}$中解出 $\boldsymbol{x}$.

虽然 LU 分解法把一个方程组的求解转化为两个方程组的求解,但由于现在每个方程组的系数矩阵都是三角矩阵,因此计算起来的速度是很快的.

这样如果使用 LU 分解法求解线性方程组,就包含两个步骤:分解与回代.对应的 Mathematica 命令分别为 **LUDecomposition** 与 **LUBackSubstitution**.

- **LUDecomposition[矩阵]**求出矩阵的 LU 分解.
- **LUBackSubstitution[数据, b]** 利用 LUDecomposition[矩阵]输出的结果求解方程组**矩阵.x = b.**

LUDecomposition 的输出为数据,由三部分组成:(i) 矩阵 L 与 U 被压缩到一个矩阵中,(ii) 一个置换向量,以及 (iii) 矩阵的 L^∞ 条件数.把数据送给 LUBackSubstitution 就可以求解出线性方程组.

置换向量对行进行重新安排,以保证矩阵具有最大的数值稳定性.本书中我们对条件数不加讨论.

LUDecomposition 与 LUBackSubstitution 不能用于求解具有无穷组解的线性方程组.

例 17　为了用 LU 分解法求解线性方程组 $2x + y + z = 7, x - 4y + 3z = 2, 3x + 2y + 2z = 13$,我们首先对系数矩阵进行矩阵分解.

$$\mathbf{a} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & -4 & 3 \\ 3 & 2 & 2 \end{bmatrix};\quad \mathbf{b} = \begin{bmatrix} 7 \\ 2 \\ 13 \end{bmatrix};$$

data = LUDecomposition[a]

$\{\{\{1, -4, 3\}, \{2, 9, -5\}, \{3, \frac{14}{9}, \frac{7}{9}\}\}, \{2, 1, 3\}, 1\}$

LUBackSubstitution[data, b]

{{1}, {2}, {3}}

在例 18 与例 19 中演示了 LUDecomposition 返回数据的结构.

例 18　$\mathbf{m} = \begin{bmatrix} 2 & 3 & 4 \\ 4 & 11 & 14 \\ 6 & 29 & 43 \end{bmatrix};$

{lu, p, cond} = LUDecomposition[m]

{{{2,3,4}, {2,5,6}, {3,4,7}}, {1,2,3}, 1}

在这个例子中,没有进行行置换,因为这时的置换向量 p 就是{1,2,3}.

LUDecomposition[m]的第一部分是以压缩形式给出的两个矩阵.因为我们已知 LU 的形式为 $\begin{bmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ x & x & 1 \end{bmatrix}\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{bmatrix}$,所以指定矩阵 L, U,只需要指定九个元素(用 x 表示).在 LUDecomposition[m]的第一部分就是用单个矩阵的形式给出这九个数.

lu //MatrixForm

$$\begin{bmatrix} 2 & 3 & 4 \\ 2 & 5 & 6 \\ 3 & 4 & 7 \end{bmatrix}$$

这些数虽然是组合在一个矩阵中，但每个数所在的位置却是正确的，因此

$$lu = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \\ 0 & 5 & 6 \\ 0 & 0 & 7 \end{bmatrix}.$$

例 19　$\mathbf{m} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & -4 & 3 \\ 3 & 2 & 2 \end{bmatrix}$;

{lu, p, cond} = LUDecomposition[m]

$\{\{\{1, -4, 3\}, \{2, 9, -5\}, \{3, \frac{14}{9}, \frac{7}{9}\}\}, \{2, 1, 3\}, 1\}$

lu//MatrixForm

$$\begin{bmatrix} 1 & -4 & 3 \\ 2 & 9 & -5 \\ 3 & \frac{14}{9} & \frac{7}{9} \end{bmatrix}$$

如果仍然按前面例子同样的步骤处理，就会得到

$$\mathbf{l} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & \frac{14}{9} & 1 \end{bmatrix};$$

$$\mathbf{u} = \begin{bmatrix} 1 & -4 & 3 \\ 0 & 9 & -5 \\ 0 & 0 & \frac{7}{9} \end{bmatrix};$$

但这时 l 与 u 乘起来并不等于原来的矩阵.

l.u//MatrixForm

$$\begin{bmatrix} 1 & -4 & 3 \\ 2 & 1 & 1 \\ 3 & 2 & 2 \end{bmatrix}$$

实际上这时置换向量为 p = {2, 1, 3}，表明矩阵中第一行与第二行交换了顺序. 如果这时交换 l 中相应的行，再乘上 u 就会得到原来的矩阵.

$$\mathbf{l} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 3 & \frac{14}{9} & 1 \end{bmatrix};$$

l.u//MatrixForm

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & -4 & 3 \\ 3 & 2 & 2 \end{bmatrix}$$

重构出 L 与 U 矩阵的更方便的方法是利用 **LUMatrices** 命令，这条命令包含在软件包 **LinearAlgebra`MatrixManipulation`** 中.

■ **LUMatrices[lu]**返回一个列表，形式为{l, u}，由对应于 **lu** 的矩阵在 LU 分解中被置换后的上三角与下三角矩阵组成，其中 **lu** 为 LUDecomposition[矩阵]的第一个元素.

例 20　**<<LinearAlgebra`MatrixManipulation`**

```
m = {{2, 1, 1}, {1, -4, 3}, {3, 2, 2}};
{lu, p, cond} = LUDecomposition[m]
```

$\{\{\{1, -4, 3\}, \{2, 9, -5\}, \{3, \frac{14}{9}, \frac{7}{9}\}\}, \{2, 1, 3\}, 1\}$

```
LUMatrices[lu]
```

$\{\{\{1, 0, 0\}, \{2, 1, 0\}, \{3, \frac{14}{9}, 1\}\}, \{\{1, -4, 3\}, \{0, 9, -5\}, \{0, 0, \frac{7}{9}\}\}\}$

```
l = LUMatrices[lu][[1]];
u = LUMatrices[lu][[2]];
l[[p]]//MatrixForm
```

← l[[p]]根据向量 p 对矩阵 l 的行进行置换.

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 3 & \frac{14}{9} & 1 \end{bmatrix}$$

```
u//MatrixForm
```

$$\begin{bmatrix} 1 & -4 & 3 \\ 0 & 9 & -5 \\ 0 & 0 & \frac{7}{9} \end{bmatrix}$$

```
l[[p]].u//MatrixForm
```

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & -4 & 3 \\ 3 & 2 & 2 \end{bmatrix}$$

习 题 解 答

12.23 由 (1, 2, 1, 2, 1),(1, 3, 2, 4, 2) 与 (1, 4, 3, 6, 3) 张成向量集合 S,描述这个向量集合的结构.

解

```
a = {1, 2, 1, 2, 1};
b = {1, 3, 2, 4, 2};
c = {1, 4, 3, 6, 3};
m = {a, b, c};
m //MatrixForm
```

$$\begin{bmatrix} 1 & 2 & 1 & 2 & 1 \\ 1 & 3 & 2 & 4 & 2 \\ 1 & 4 & 3 & 6 & 3 \end{bmatrix}$$

> 由给定向量作为行生成矩阵 m. m 的行空间就是由 a, b, c 张成的空间. 然后把这个矩阵转化为简化行梯形形式. 非零行就成为行空间的一组基. 在 S 中每个向量都是这组基向量的线性组合.

```
rref = RowReduce[m];
rref //MatrixForm
```

$$\begin{bmatrix} 1 & 0 & -1 & -2 & -1 \\ 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
s * rref[[1]] + t * rref[[2]]
```

{s, t, - s + t, - 2s + 2t, - s + t}　←在 S 中每个向量都具有这种形式.

12.24 在线性代数中有一个定理,说的是在 A 的行空间中每个向量都正交于 A 的零空间中的每个向量. 请对下述矩阵验证这个定理:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}.$$

解 ☞

只要证明行向量中每个基向量正交于零空间中的每个基向量就可以了.

```
a = Partition[Range[25], 5];
a //MatrixForm
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

```
rowspacebasis = RowReduce[a];
rowspacebasis //MatrixForm
```

$$\begin{bmatrix} 1 & 0 & -1 & -2 & -3 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

```
nullspacebasis = NullSpace[a];
nullspacebasis //MatrixForm
```

$$\begin{bmatrix} 3 & -4 & 0 & 0 & 1 \\ 2 & -3 & 0 & 1 & 0 \\ 1 & -2 & 1 & 0 & 0 \end{bmatrix}$$

```
rowspacebasis.Transpose[nullspacebasis]//MatrixForm
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

12.25 构造一个 5 × 5 阶由随机数字组成的矩阵 **a**, 再构造一个 5 × 1 阶由随机数字组成的矩阵 **b**, 然后用 **LinearSolve** 求解线性方程组 **a.x = b**, 并验证解的正确性.

解 ☞

```
a = Table[Random[Integer, {0,9}], {i,1,5}, {j,1,5}];
a //MatrixForm
```

$$\begin{bmatrix} 0 & 9 & 5 & 0 & 2 \\ 7 & 7 & 5 & 9 & 3 \\ 0 & 6 & 6 & 2 & 6 \\ 5 & 8 & 4 & 3 & 9 \\ 3 & 0 & 8 & 1 & 5 \end{bmatrix}$$

```
Det[a] == 0
False
```
←由于行列式不等于0,所以这个方程组有惟一解.

```
b = Table[Random[Integer, {0,9}], {i,1,5}];
b//MatrixForm
```

$$\begin{bmatrix} 1 \\ 2 \\ 4 \\ 6 \\ 1 \end{bmatrix}$$

```
x = LinearSolve[a, b]
```

$$\left\{\left\{-\frac{58}{217}\right\},\left\{\frac{65}{651}\right\},\left\{-\frac{187}{651}\right\},\left\{\frac{111}{434}\right\},\left\{\frac{143}{186}\right\}\right\}$$

```
a.x == b
True
```

12.26 求出线性方程组$\begin{cases} w+2x+3y+3z=9, \\ 2w+x+2y+5z=10, \\ 2w+2x+y+2z=7, \\ 2w-x-3y+z=-1 \end{cases}$的一般解.

解 ☞

$$a = \begin{bmatrix} 1 & 2 & 3 & 3 \\ 2 & 1 & 2 & 5 \\ 2 & 2 & 1 & 2 \\ 2 & -1 & -3 & 1 \end{bmatrix};\ b = \begin{bmatrix} 9 \\ 10 \\ 7 \\ -1 \end{bmatrix};$$

```
Det[a]
0
```
←由于行列式为零,因此方程组只可能没有解或者有无穷组解.

```
nullspacebasis = NullSpace[a]
{{-13, 11, -10, 7}}
```
←由于零空间非空,所以方程组有无穷组解.

```
particular = LinearSolve[a, b]
```

$$\left\{\left\{\frac{20}{7}\right\},\left\{-\frac{4}{7}\right\},\left\{\frac{17}{7}\right\},\{0\}\right\}$$

```
generalsolution = t * Flatten[nullspacebasis] + Flatten[particular]
```

$$\left\{\frac{20}{7}-13t,\ -\frac{4}{7}+11t,\ \frac{17}{7}-10t,\ 7t\right\}$$

12.27 求出线性方程组$\begin{cases} w+2x+3y+3z=9, \\ 3w+4x+4y+5z=16, \\ 2w+2x+y+2z=7, \\ 4w+6x+7y+8z=25 \end{cases}$的一般解.

解 ☞

$$a = \begin{bmatrix} 1 & 2 & 3 & 3 \\ 3 & 4 & 4 & 5 \\ 2 & 2 & 1 & 2 \\ 4 & 6 & 7 & 8 \end{bmatrix};\ b = \begin{bmatrix} 9 \\ 16 \\ 7 \\ 45 \end{bmatrix};$$

```
Det[a]
0
nullspacebasis = NullSpace[a]
{{1, -2, 0, 1}, {4, -5, 2, 0}}
particular = LinearSolve[a, b]
```

$\{\{-2\}, \{\frac{11}{2}\}, \{0\}, \{0\}\}$

```
generalsolution = s * nullspacebasis[[1]] +
                  t * nullspacebasis[[2]] + Flatten[particular]
```

$\{-2 + s + 4t, \frac{11}{2} - 2s - 5t, 2t, s\}$

12.28 令 A 为 7×7 阶三对角矩阵，在主对角线上的元素都是 3，与主对角线相邻的次对角线上元素都是 -1. 令 e_i 表示第 i 个分量为 1，其它分量都是 0 的七维向量. 求解方程组 $Ax = e_i, i = 1, \cdots, 7$.

解 ☞

```
a = Table[If[Abs[i - j] == 1, -1, If[i == j, 3, 0]], {i,1,7},{j,1,7}];
a//MatrixForm
```

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 \end{bmatrix}$$

```
ludata = LUDecomposition[a];
b = Table[KroneckerDelta[i, j], {i,1,7}, {j,1,7}];
LUBackSubstitution[ludata, b]//TableForm
```

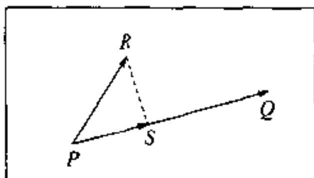
$$\begin{matrix} \frac{377}{987} & \frac{48}{329} & \frac{55}{987} & \frac{1}{47} & \frac{8}{987} & \frac{1}{329} & \frac{1}{987} \\ \frac{48}{329} & \frac{144}{329} & \frac{55}{329} & \frac{3}{47} & \frac{8}{329} & \frac{3}{329} & \frac{1}{329} \\ \frac{55}{987} & \frac{55}{329} & \frac{440}{987} & \frac{8}{47} & \frac{64}{987} & \frac{8}{329} & \frac{8}{987} \\ \frac{1}{47} & \frac{3}{47} & \frac{8}{47} & \frac{21}{47} & \frac{8}{47} & \frac{3}{47} & \frac{1}{47} \\ \frac{8}{987} & \frac{8}{329} & \frac{64}{987} & \frac{8}{47} & \frac{440}{987} & \frac{55}{329} & \frac{55}{987} \\ \frac{1}{329} & \frac{3}{329} & \frac{8}{329} & \frac{3}{47} & \frac{55}{329} & \frac{144}{329} & \frac{48}{329} \\ \frac{1}{987} & \frac{1}{329} & \frac{8}{987} & \frac{1}{47} & \frac{55}{987} & \frac{48}{329} & \frac{377}{987} \end{matrix}$$

> 当 $i = j$ 时，KroneckerDelta[i, j] = 1，否则等于 0.

这个表格的第 i 列表示方程组 $Ax = e_i$ 的解.

12.5 正交性

给定两个向量,如果它们的内积等于零,就称它们是正交的(垂直的).正交向量具有许多有用的性质,从而使得对它们的处理非常方便.例如,若 $\boldsymbol{u}$ 与 $\boldsymbol{v}$ 为正交向量,它们就满足(广义的)勾股定理:$\|\boldsymbol{u}+\boldsymbol{v}\|^2=\|\boldsymbol{u}\|^2+\|\boldsymbol{v}\|^2$.



通过正交性也可以定义投影的概念.在$\mathbb{R}^2$ 中很容易看出投影的意义.如果 $\boldsymbol{a}=\overrightarrow{PQ}$与 $\boldsymbol{b}=\overrightarrow{PR}$为具有同样起点 P 的两个向量,而且如果 S 为从 R 到$\overrightarrow{PQ}$作垂线的垂足,那么 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 上的投影就是向量$\overrightarrow{PS}$.这个向量通常用 $\mathrm{proj}_{\mathbf{a}}\boldsymbol{b}$ 表示.

可以利用 Mathematica 命令 **Projection** 计算投影向量,这条命令包含在软件包 **LinearAlgebra`Orthogonalization`** 中.

■ **Projection[向量 1, 向量 2]** 给出向量 1 在向量 2 上的投影.

例 21 下述命令序列可以计算 (1, 2, 3) 在 (−2, 3, −1) 上的投影.

```
<<LinearAlgebra`Orthogonalization`
a = {1, 2, 3};
b = {-2, 3, -1};
Projection[a, b]
```

$\left\{-\frac{1}{7},\frac{3}{14},-\frac{1}{14}\right\}$

正交性与所考虑空间中内积的定义有关.默认情况下,Mathematica 使用线性代数中的欧几里得内积(点积).然而,可以通过包含 **InnerProduct** 选项来改变这一点.

- **InnerProduct →函数**,其中函数为一个纯粹函数,定义所考虑空间中的内积.(关于纯粹函数的讨论,请见附录.)

可以证明,如果 c_1,c_2,c_3 为正实数,那么$\langle\boldsymbol{a},\boldsymbol{b}\rangle=c_1a_1b_1+c_2a_2b_2+c_3a_3b_3$ 定义了 $\mathbb{R}^3$ 空间中的内积.为了利用这种内积计算 (1, 2, 3) 在 (−2, 3, −1) 上的投影,我们就必须定义适当地描述这种内积的纯粹函数.为此,首先计算向量 $\boldsymbol{a}*\boldsymbol{b}$,然后计算所得结果与 $c=(c_1, c_2, c_3)$的内积.下面的例子演示了这种方法.

例 22 利用内积$\langle\mathbf{a},\mathbf{b}\rangle=2a_1b_1+3a_2b_2+4a_3b_3$ 可以计算 (1, 2, 3) 在 (−2, 3, −1) 上的投影.

```
<<LinearAlgebra`Orthogonalization`
a = {1, 2, 3};
b = {-2, 3, -1};
Projection[a, b, InnerProduct →({2,3,4}.{#1 * #2}&)]
```

$\left\{-\frac{4}{39},\frac{2}{13},-\frac{2}{39}\right\}$

例 23　在函数空间中常用的内积通常定义为$\langle f, g\rangle = \int_{-1}^{1} f(x)g(x)\mathrm{d}x$．请利用这种定义计算 x^2 在 x^3+1 上的投影．

```
<<LinearAlgebra`Orthogonalization`
a = x^2;
b = x^3 + 1;
Projection[a, b, InnerProduct →(Integrate[#1 * #2, {x, -1, 1}]&)]
```

$\frac{7}{24}(1 + x^3)$

由定义，所谓有限维向量空间，就是具有有限基的向量空间．然而，除了只包含零向量的平凡向量空间外，其它向量空间都可以具有无穷组不同的基．

对于向量空间而言，最方便的基是标准正交基．Gram-Schmidt 正交化过程就是一种把普通基转化标准正交基的方法．

■ **Normalize[向量]** 把向量转化为单位向量．

■ **GramSchmidt[向量列表]** 给出由向量列表张成空间的一组标准正交向量组．

- **Normalized →False** 选项可以包含在 GramSchmidt 命令中，从而使得这种命令生成正交向量组，但不必是标准正交向量组．
- **InnerProduct →函数** 也可以用在这条命令中，定义不同于默认内积（点积）的内积．

与 Projection 一样，Normalize 与 GramSchmidt 也是包含在 LinearAlgebra`Orthogonalization`软件包中．

例 24　为了相应于欧几里得内积把 (3, 4, 12) 单位化，输入

```
<<LinearAlgebra`Orthogonalization`
Normalize[{3, 4, 12}]
```

$\left\{\frac{3}{13}, \frac{4}{13}, \frac{12}{13}\right\}$

为了相应于内积$\langle a, b\rangle = 2a_1b_1 + 3a_2b_2 + 4a_3b_3$把这个向量标准化，输入

```
Normalize[{3, 4, 12}, InnerProduct →({2,3,4}·{#1 * #2}&)]
```

$\left\{\sqrt{\frac{3}{214}}, 2\sqrt{\frac{2}{321}}, 2\sqrt{\frac{6}{107}}\right\}$

例 25　给出由 (1, 1, 1, 0, 0), (0, 1, 1, 1, 0) 与 (0, 0, 1, 1, 1) 张成空间的一组标准正交基，并验证所得结果的正确性．

```
<<LinearAlgebra`Orthogonalization`
v = {{1,1,1,0,0}, {0,1,1,1,0}, {0,0,1,1,1}};
w = GramSchmidt[v]
```

$\left\{\left\{\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0, 0\right\}, \left\{-\frac{2}{\sqrt{15}}, \frac{1}{\sqrt{15}}, \frac{1}{\sqrt{15}}, \sqrt{\frac{3}{5}}, 0\right\},\right.$

$\left.\left\{\frac{1}{2\sqrt{10}}, -\frac{3}{2\sqrt{10}}, \frac{1}{\sqrt{10}}, \frac{1}{2\sqrt{10}}, \frac{\sqrt{\frac{5}{2}}}{2}\right\}\right\}$

为了验证所得结果的正确性，需要计算六个内积．

```
w[[1]].w[[1]]
```

1

```
w[[2]].w[[2]]
1
w[[3]].w[[3]]
1                        ←这说明向量都具有单位长度.
w[[1]].w[[2]]
0
w[[2]].w[[3]]
0
w[[3]].w[[1]]
0                        ←这说明不同的向量是正交的.
```

习题解答

12.29 计算向量 (1, 2, 3, 4, 5) 相应于下述内积的模:(a) 欧几里得内积;(b) $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 2u_1v_1 + 3u_2v_2 + u_3v_3 + 3u_4v_4 + 2u_5v_5$.

解 ☞

(a)

```
u = {1,2,3,4,5};
norm = √u.u
√55
```

(b)

```
c = {2,3,1,3,2};
norm = √c.(u*u)
11
```

12.30 计算向量 (a, b, c) 在每个坐标轴上的投影.

解 ☞

```
<<LinearAlgebra`Orthogonalization`
v = {a, b, c};
Projection[v, {1, 0, 0}]
{a, 0, 0}
Projection[v, {0, 1, 0}]
{0, b, 0}
Projection[v, {0, 0, 1}]
{0, 0, c}
```

12.31 求出与 (1, −2, 2, −3) 有同样方向的单位向量.

解 ☞

```
<<LinearAlgebra`Orthogonalization`
Normalize[{1, -2, 2, -3}]
```

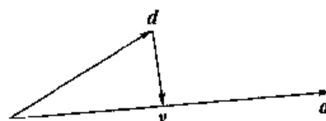
$\left\{\frac{1}{3\sqrt{2}}, -\frac{\sqrt{2}}{3}, \frac{\sqrt{2}}{3}, -\frac{1}{\sqrt{2}}\right\}$

12.32 如果 $\boldsymbol{a} = (1, 2, 3), \boldsymbol{b} = (1, -2, 5)$,请计算在下图中所示向量 $\boldsymbol{v}$ 的长度.

解 ☞

由于 $\boldsymbol{b} + \boldsymbol{v} = \text{proj}_a\boldsymbol{b}$,因此 $\boldsymbol{v} = \text{proj}_a\boldsymbol{b} - \boldsymbol{b}$.

```
<<LinearAlgebra`Orthogonalization`
a = {1, 2, 3}; b = {1, -2, 5};
```




```
v = b - Projection[b,a];
norm = Sqrt[v.v]
```

$\sqrt{\frac{138}{7}}$

12.33 求出由 (1, 2, 1, 3), (2, 2, 2, 2), (1, −1, 1, −1) 与 (3, 4, 3, 5) 张成空间的正交基.

解 ☞

由于需要的只是正交向量,因此在 **GramSchmidt** 命令中使用 **Normalized →False** 选项.

```
<<LinearAlgebra`Orthogonalization`
v1 = {1, 2, 1, 3};
v2 = {2, 2, 2, 2};
v3 = {1, -1, 1, -1};
v4 = {3, 4, 3, 5};
v = {v1, v2, v3, v4};
w = GramSchmidt[v, Normalized →False]
```

$\left\{\{1,2,1,3\},\left\{\frac{16}{15},\frac{2}{15},\frac{16}{15},-\frac{4}{5}\right\}\right.$
$\left.\left\{\frac{2}{11},-\frac{8}{11},\frac{2}{11},\frac{4}{11}\right\},\{0,0,0,0\}\right\}$

> 由于向量组 v 是线性相关的,因此 GramSchmidt 只能给出三个正交向量,这里的{0,0,0,0}可被忽略.

12.34 给定由不超过五次的全体多项式构成的线性空间 P_5,求出这个空间相应于内积 $\langle p, q\rangle = \int_{-1}^{1} p(x)q(x)\mathrm{d}x$ 的正交基.

解 ☞

P_5 有一组基就是 v = $\{1, x, x^2, x^3, x^4, x^5\}$.这组向量构成线性无关组,并张成空间 P_5.

```
<<LinearAlgebra`Orthogonalization`
v = {1, x, x^2, x^3, x^4, x^5};
    GramSchmidt[v, InnerProduct →(Integrate[#1 #2, {x, -1, 1}]&),
              Normalized →False]//Expand
```

$\left\{1, x, -\frac{1}{3}+x^2, -\frac{3x}{5}+x^3, \frac{3}{35}-\frac{6x^2}{7}+x^4, \frac{5x}{21}-\frac{10x^3}{9}+x^5\right\}$

注意　在这道题中得到的函数就是勒让德多项式,并对它们进行了规范化处理,因此首项系数都是 1. 在 Mathematica 中用 **LegendreP** 表示这些多项式.

```
normalizedlegendre :=
  LegendreP[n,x]/Coefficient[LegendreP[n,x],x,n]
Table[{n, normalizedlegendre}, {n, 0, 5}]//Expand//TableForm
```

```
0    1
1    x
```

2　$-\frac{1}{3}+x^2$

3　$-\frac{3}{5}x+x^3$

4　$\frac{3}{35}-\frac{6x^2}{7}+x^4$

5　$\frac{5x}{21}-\frac{10x^3}{9}+x^5$

12.6 特征值与特征向量

我们称数 λ 为方阵 A 的一个特征值,是指存在非零向量 x,使得 $Ax = \lambda x$,这时 x 称为相应于这个特征值的特征向量.虽然特征值概念在线性代数中是非常重要的,但是当矩阵维数很大时,特征值以及对应的特征向量的计算是非常困难的.

确定矩阵特征值的一种方法就是求解对应的特征方程 $\det(A - \lambda I) = 0$.一旦确定了特征值,那么通过求解齐次线性方程组就可以确定对应的特征向量.

例 26

```
a = {{4, 1, -1}, {2, 5, -2}, {1, 1, 2}};
len = Length[a];
Solve[Det[a - λ IdentityMatrix[len]] == 0, λ]
{{λ → 3}, {λ → 3}, {λ → 5}}
```

所以特征值为 3(重数为 2)与 5.为了求出相应的特征向量,我们看一下 $A - \lambda I$ 的零空间.

```
NullSpace[a - 3 IdentityMatrix[len]]
{{1,0,1},{-1,1,0}}
NullSpace[a - 5 IdentityMatrix[len]]
{{1,2,1}}
```

当然,正如我们所期望的那样,在 Mathematica 中包含自动计算特征值、特征向量以及其它一些与特征值和特征向量有关的项的命令.

- **CharacteristicPolynomial[矩阵, 变量]** 给出矩阵的特征多项式,并把结果表示为指定变量的多项式.
- **Eigenvalues[矩阵]** 给出矩阵的特征值列表.
- **Eigenvectors[矩阵]** 给出矩阵的特征向量列表.
- **Eigensystem[矩阵]** 返回相应于矩阵的形式为{特征值,特征向量}列表.

例 27

```
a = {{4, 1, -1}, {2, 5, -2}, {1, 1, 2}};
CharacteristicPolynomial[a,x]
45 - 39x + 11x^2 - x^3
Eigenvalues[a]
{3,3,5}
Eigenvectors[a]
{{1,0,1},{-1,1,0},{1,2,1}}
```

```
Eigensystem[a]
{{3, 3, 5}, {{1, 0, 1}, {-1, 1, 0}, {1, 2, 1}}}
```

如果矩阵中的各个元素是以准确形式(即不是数值形式)给出的，那么 Mathematica 就会尽力给出矩阵准确的特征值与特征向量. 如果在矩阵中有的元素是以数值形式给出的，那么 Mathematica 就会给出数值近似结果. 另外，也可以利用 **N[矩阵]** 作为 CharacteristicPolynomial、Eigenvalues 与 Eigenvectors 的参数，以强迫 Mathematica 利用数值算法给出所期望的结果. 如果希望得到高的精度，那么可以使用 **N[矩阵, k]** 以得到 k 位有效数字.

注意 由于数值计算特征值所用的算法与精确算法不同，因此可能会导致所得的结果顺序不同. 而且，由于特征向量并不是惟一确定的，因此数值特征向量也可能与前面得到的不同，但它们都是合法的特征向量.

例 28 $a = \begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix};$

```
Eigenvalues[a]
{2 - √2, 2 + √2}
Eigenvectors[a]
{{-1 - √2, 1}, {-1 + √2, 1}}
Eigenvalues[N[a]]
{3.41421, 0.585786}
Eigenvectors[N[a]]
{{-0.382683, -0.92388}, {-0.92388, 0.382683}}
Eigenvalues[N[a, 20]]
{3.4142135623730950488, 0.5857864376269049512}
Eigenvectors[N[a,20]]
{{-0.38268343236508977173, -0.92387953251128675613},
 {-0.92387953251128675613, 0.38268343236508977173}}
```

习 题 解 答

12.35 方阵 A 由前 25 个相邻整数构成，那么它的特征多项式是什么呢?

解 ☞

```
a = Partition[Range[25], 5];
a//MatrixForm
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$$

```
CharacteristicPolynomial[a,x]
```

$250\ x^3 + 65\ x^4 - x^5$

12.36 考虑 5 × 5 阶三对角矩阵，其主对角线上的元素都是 4，与主对角线相邻的次对角线上元素都是 1. 用一种明了的方式给出这个矩阵的特征值以及对应的特征向量.

解 ☞

```
m = Table[If[Abs[i - j] == 1, 1, If[i == j, 4, 0]], {i,1,5}, {j,1,5}];
m//MatrixForm
```

$$\begin{bmatrix} 4 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

```
data = Eigensystem[m]
```

$\{\{3, 4, 5, 4-\sqrt{3}, 4+\sqrt{3}\}, \{\{-1,1,0,-1,1\}, \{1,0,-1,0,1\}, \{-1,-1,0,1,1\}, \{1,-\sqrt{3},2,-\sqrt{3},1\}, \{1,\sqrt{3},2,\sqrt{3},1\}\}\}$

```
Do[Print["eigenvalue #",k, "is", data[[1,k]],
  "with corresponding eigenvector: ", data[[2,k]]], {k,1,5}]
```

eigenvalue #1 is 3 with corresponding eigenvector: $\{-1,1,0,-1,1\}$
eigenvalue #2 is 4 with corresponding eigenvector: $\{1,0,-1,0,1\}$
eigenvalue #3 is 5 with corresponding eigenvector: $\{-1,-1,0,1,1\}$

eigenvalue #5 is $4-\sqrt{3}$ with corresponding eigenvector: $\{1,-\sqrt{3},2,-\sqrt{3},1\}$
eigenvalue #5 is $4+\sqrt{3}$ with corresponding eigenvector: $\{1,\sqrt{3},2,\sqrt{3},1\}$

12.37 在线性代数中有一个重要的定理，即 Cayley-Hamilton 定理，这个定理指的是矩阵满足它的特征多项式. 对于下述矩阵验证 Cayley-Hamilton 定理：

$$A = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 3 & -1 & 3 & -1 \\ 2 & 5 & 7 & 1 \\ 1 & 2 & 3 & 6 \end{bmatrix}.$$

解 ☞

$$a = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 3 & -1 & 3 & -1 \\ 2 & 5 & 7 & 1 \\ 1 & 2 & 3 & 6 \end{bmatrix};$$

```
CharacteristicPolynomial[a,x]
```

$-196 + 161x + 15x^2 - 13x^3 + x^4$

```
-196 IdentityMatrix[4] + 161 a + 15 MatrixPower[a,2]
  -13 MatrixPower[a,3] + MatrixPower[a,4]//MatrixForm
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

12.38 给出 10×10 阶希耳伯特矩阵 $h_{i,j} = \dfrac{1}{i+j-1}$ 的近似特征值.

解 ☞

```
n = 10;
hilbert = Table[1/(i + j - 1), {i,1,n}, {j,1,n}];
hilbert//MatrixForm
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} \\ \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} \\ \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} \\ \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} & \frac{1}{19} \end{bmatrix}$$

Eigenvalues[N[hilbert]]

{1.75192, 0.34293, 0.0357418, 0.00253089, 0.00012875, 4.72969×10^{-6}, 1.22897×10^{-7}, 2.14744×10^{-9}, 2.26675×10^{-11}, 1.09325×10^{-13}}

12.39 给出元素为如下表示的 10×10 阶矩阵 A 的近似特征值：

$$a_{i,j} = \begin{cases} i+j-1, \text{如果 } i+j \leqslant 11, \\ 21-i-j, \text{如果 } i+j > 11. \end{cases}$$

解 ☞

```
f[i_, j_] := i + j - 1/;i + j ≤11
f[i_, j_] := 21 - i - j/;i + j >11
a = Array[f, {10,10}];
a//MatrixForm
```

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 \\ 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 & 8 \\ 4 & 5 & 6 & 7 & 8 & 9 & 10 & 9 & 8 & 7 \\ 5 & 6 & 7 & 8 & 9 & 10 & 9 & 8 & 7 & 6 \\ 6 & 7 & 8 & 9 & 10 & 9 & 8 & 7 & 6 & 5 \\ 7 & 8 & 9 & 10 & 9 & 8 & 7 & 6 & 5 & 4 \\ 8 & 9 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 \\ 9 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

Eigenvalues[N[a]]

{67.8404, −20.4317, 4.45599, −2.42592, 1.39587, −1., 0.756101, −0.629808, 0.55164, −0.512543}

12.7 对角化与约当标准型

给定 $n \times n$ 阶矩阵 A，如果存在矩阵 P，使得 $A = PDP^{-1}$，其中 D 为对角矩阵，那么称

A 为可对角化的矩阵.

并不是每个矩阵都是可对角化的.然而,可以证明,如果 A 的特征向量构成完全集,即 A 有 n 个线性无关的特征向量,那么 A 是可对角化的.这时矩阵 P 的各列元素就是 A 的特征向量,对角矩阵 D 的主对角元素就是相应的特征值.

例 29 a = $\begin{bmatrix} 18 & -51 & 27 & -15 \\ 8 & -24 & 14 & -8 \\ 15 & -48 & 28 & -15 \\ 15 & -47 & 25 & -12 \end{bmatrix}$;

```
Eigenvalues[a]
{1,2,3,4}            ←由于特征值不同,所有对应的特征向量是线性无关的.
Eigenvectors[a]
{{3,2,3,2},{0,1,3,2},{1,0,0,1},{3,1,2,3}}
p = Transpose[ % ]       ←转置使得特征向量成为矩阵的列.
{{3,0,1,3},{2,1,0,1},{3,3,0,2},{2,2,1,3}}
d = DiagonalMatrix[Eigenvalues[a]]
{{1,0,0,0},{0,2,0,0},{0,0,3,0},{0,0,0,4}}
p.d.Inverse[p]//MatrixForm
```

$\begin{bmatrix} 18 & -51 & 27 & -15 \\ 8 & -24 & 14 & -8 \\ 15 & -48 & 28 & -15 \\ 15 & -47 & 25 & -12 \end{bmatrix}$ ← p.d.Inverse[p] = a.

总之,

```
MatrixForm[a] == MatrixForm[p].MatrixForm[d].MatrixForm[Inverse[p]]
```

$$\begin{bmatrix} 18 & -51 & 27 & -15 \\ 8 & -24 & 14 & -8 \\ 15 & -48 & 28 & -15 \\ 15 & -47 & 25 & -12 \end{bmatrix} == \begin{bmatrix} 3 & 0 & 1 & 3 \\ 2 & 1 & 0 & 1 \\ 3 & 3 & 0 & 2 \\ 2 & 2 & 1 & 3 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} . \begin{bmatrix} -1 & 4 & -2 & 1 \\ -1 & 2 & -1 & 1 \\ -5 & 15 & -9 & 6 \\ 3 & -9 & 5 & -3 \end{bmatrix}$$

前面已指出,并不是每个矩阵都是可对角化的.然而,存在一种形式,名为约当标准型,每个矩阵都可以变换到这种形式上.

一个约当块就是一个方阵,主对角线上元素相同,主对角线的元素都是 1,其它元素都是零,即

$$\begin{bmatrix} \lambda & 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & 0 & \cdots & 0 \\ 0 & 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \lambda \end{bmatrix}.$$

如果 A 为 $n \times n$ 阶矩阵,那么存在矩阵 Q,使得 $A = QJQ^{-1}$,其中

$$J = \begin{bmatrix} J_1 & 0 & 0 & \cdots & 0 \\ 0 & J_2 & 0 & \cdots & 0 \\ 0 & 0 & J_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & J_k \end{bmatrix},$$

J_i 为约当块.在不同块中的特征值可以不同.对于给定的一个特征值,它所对应的不同约当块

的数目等于对应于这个特征值的线性无关的特征向量数.

例 30　$a = \begin{bmatrix} 5 & 4 & 3 \\ -1 & 0 & -3 \\ 1 & -2 & 1 \end{bmatrix}$;

```
Eigensystem[a]
{{-2,4,4}, {{-1,1,1},{1,-1,1},{0,0,0}}}
```

所以特征值为 −2 与 4,分别对应的特征向量为 (−1, 1, 1) 与 (1, −1, 1). 向量 {0, 0, 0} 不是特征向量,之所以出现它,就是因为不能找到第二个线性无关的特征向量. 为了构造出矩阵 Q,标准的做法是找到向量 x 使得 $(A - 4I)x = (1, -1, 1)$. x 称为广义特征向量.

```
LinearSolve[a - 4 IdentityMatrix[3], {1, -1, 1}]
{1, 0, 0}
```

这样就可以构造出矩阵 Q 与 J.

$q = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$;　　q 的列由 a 的特征向量与广义特征向量构成.

$j = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 0 & 4 \end{bmatrix}$;

```
q.j.Inverse[q]//MatrixForm
```

$\begin{bmatrix} 5 & 4 & 3 \\ -1 & 0 & -3 \\ 1 & -2 & 1 \end{bmatrix}$

■ **JordanDecomposition[矩阵]** 计算出矩阵的约当标准型. 所得结果形式为 {q, j}, 其中 q 与 j 分别对应于上面描述的矩阵 Q 与 J.

例 31　$a = \begin{bmatrix} 5 & 4 & 3 \\ -1 & 0 & -3 \\ 1 & -2 & 1 \end{bmatrix}$;

```
{q, j} = JordanDecomposition[a];
q //MatrixForm
```

$\begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$

```
j //MatrixForm
```

$\begin{bmatrix} -2 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 0 & 4 \end{bmatrix}$　　←当然这与在例 30 中得到的结果一致.

例 32　$a = \begin{bmatrix} 18 & -51 & 27 & -15 \\ 8 & -24 & 14 & -8 \\ 15 & -48 & 28 & -15 \\ 15 & -47 & 25 & -12 \end{bmatrix}$;

```
{q, j} = JordanDecomposition[a];
q //MatrixForm
```

$$\begin{bmatrix} 3 & 0 & 1 & 3 \\ 2 & 1 & 0 & 1 \\ 3 & 3 & 0 & 2 \\ 2 & 2 & 1 & 3 \end{bmatrix}$$

```
j //MatrixForm
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

> 在这个例子中,约当矩阵简化为对角阵,这是因为矩阵 a 具有完全集的特征向量

在 Mathematica 中还有可以进行 QR 分解与 Schur 分解等其它分解操作的命令,但本书中在此不对它们进行讨论.相应命令的名称分别为 **QRDecomposition** 与 **SchurDecomposition**.

习 题 解 答

12.40　构造一个 5 × 5 阶矩阵,它的特征值分别为 −2, −1, 0, 1, 2, 相应的特征向量为 (1, 1, 0, 0, 0), (0, 1, 1, 0, 0), (0, 0, 1, 1, 0), (0, 0, 0, 1, 1), (1, 0, 0, 0, 1).

解 ☞

```
d = DiagonalMatrix[{-2, -1, 0, 1, 2}];
p = Transpose[{{1, 1, 0, 0, 0}, {0, 1, 1, 0, 0}, {0, 0, 1, 1, 0},
              {0, 0, 0, 1, 1}, {1, 0, 0, 0, 1}}];
a = p.d.Inverse[p];
a//MatrixForm
```

$$\begin{bmatrix} 0 & -2 & 2 & -2 & 2 \\ -\frac{1}{2} & -\frac{3}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

12.41　证明矩阵 $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ 有实特征值当且仅当 $a^2 + 4bc - 2ad + d^2 \geqslant 0$.

解 ☞

```
m = {{a,b},{c,d}};
Eigenvalues[{{a,b},{c,d}}]
```

$$\left\{\frac{1}{2}\left(a+d-\sqrt{a^2+4bc-2ad+d^2}\right), \frac{1}{2}\left(a+d+\sqrt{a^2+4bc-2ad+d^2}\right)\right\}$$

所以特征值为实数当且仅当上述根号下的表达式为非负数.

12.42　构造一个 7 × 7 阶的由随机数字组成的矩阵,说明这个矩阵特征值的和等于它的迹,特征值的积等于它的行列式.

解 ☞

```
a = Table[Random[Integer, {0,9}], {i,1,7}, {j,1,7}];
a//MatrixForm
```

$$\begin{bmatrix} 6 & 8 & 7 & 0 & 9 & 5 & 3 \\ 5 & 6 & 3 & 6 & 9 & 0 & 4 \\ 5 & 3 & 3 & 0 & 0 & 1 & 9 \\ 1 & 3 & 9 & 9 & 9 & 9 & 5 \\ 5 & 0 & 2 & 8 & 4 & 1 & 3 \\ 9 & 5 & 7 & 4 & 9 & 9 & 3 \\ 1 & 3 & 1 & 5 & 5 & 1 & 3 \end{bmatrix}$$

```
eigen = Eigenvalues[N[a]]
{31.4389, 6.08322 + 4.84653 i, 6.08322 - 4.84653 i, -6.43529,
 1.52463 + 3.69404 i, 1.52463 - 3.69404 i, -0.215269}
```

$\sum_{i=1}^{7}$ `eigen[[i]]`或者 `Sum[eigen[[i]], {i,1,7}]`

```
40. + 0. i
Tr[a]
40
```

$\prod_{i=1}^{7}$ `eigen[[i]]`或者 `Product[eigen[[i]], {i,1,7}]`

```
42072. + 0. i
Det[a]
42072
```

12.43 矩阵 P 称为正交矩阵是指它满足 $P^{T}P = I$.(即正交矩阵的逆是它的转置矩阵.)如果可以找到正交矩阵 P,使得矩阵 A 对角化,则称 A 为可正交对角化的矩阵.只有对称矩阵才可以正交对角化.(矩阵为对称的是指它满足 $A = A^{T}$.如果 A 为对称阵,那么可以证明对应于不同特征值的特征向量是正交的.)对于下述矩阵 A,求出使得它正交对角化的矩阵 P:

$$A = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{bmatrix}.$$

解 ☞

```
<<LinearAlgebra`Orthogonalization`
```

$$a = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{bmatrix};$$

> 共有五个特征值,其中两个的重数为 2.把它们分成三组特征空间,并分别应用 Gram-Schmidt 正交化过程.所需正交矩阵的列就是由 Gram-Schmidt 过程后得到的正交向量构成的.

```
{values, vectors} = Eigensystem[a]
{{1,1,2,4,4}, {{0,0,-1,0,1}, {0,0,-1,1,0},
 {-1,1,0,0,0}, {0,0,1,1,1}, {1,1,0,0,0}}}
eigenspace1 = {vectors[[1]], vectors[[2]]};
```

```
eigenspace2 = {vectors[[3]]};
eigenspace3 = {vectors[[4]], vectors[[5]]};
v1 = GramSchmidt[eigenspace1];
v2 = GramSchmidt[eigenspace2];
v3 = GramSchmidt[eigenspace3];
<<LinearAlgebra`MatrixManipulation`
orthogonalMatrix = Transpose[AppendColumns[v1,v2,v3]];
orthogonalmatrix //MatrixForm
```

$$\begin{bmatrix} 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} & 0 \end{bmatrix}$$

```
p = orthogonalmatrix;
Transpose[p].p//MatrixForm
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

← 这样做就是为了检验矩阵正交性.

```
d = DiagonalMatrix[values];
p.d.Transpose[p]//MatrixForm
```

$$\begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{bmatrix}$$

←这就又得到了初始矩阵.

```
MatrixForm[a] == MatrixForm[p].MatrixForm[d].MatrixForm[Transpose[p]]
```

$$\begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 & 2 \end{bmatrix} == \begin{bmatrix} 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & \sqrt{\frac{2}{3}} & 0 & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

$$
\begin{bmatrix}
0 & 0 & -\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\
0 & 0 & -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{6}} \\
-\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\
0 & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0
\end{bmatrix}
$$

附　　录

A.1　纯粹函数

函数是两个数集 A 与 B 之间的对应关系，这个对应关系满足：对于 A 中的每个数，在 B 中都存在惟一的数与它对应．例如，平方函数定义一个对应关系，它使得每个实数，有惟一的非负实数与它对应，这个非负实数称为原来那个实数的平方．

虽然通常采用写法 $f(x) = x^2$ 来表示函数，但我们必须清楚这里的字母 x 没有任何原来字面上的特殊含义，它只是表示定义函数中的平方过程．

虽然在 Mathematica 中我们可以用变量来定义函数，例如 f[x_] = x^2，但是变量 x 只是一个哑元，没有任何本来意义．即使我们把 x 换成 y 或 z，甚至其它符号，这也表示同样的函数．

所谓纯粹函数就是不引用特定变量进行定义的函数，这时定义中的变量依次用 #1, #2, #3 等表示．为了把纯粹函数与其它的 Mathematica 构造区分开来，在函数定义的尾部加上符号 &．一旦定义了纯粹函数，那么就可以如同其它函数一样进行处理．

例 1
```
f = #1^2 &;
f[3]
9
f[x]
x^2
f[a+b]
(a+b)^2
```

例 2
```
g = #1 #2^2 + 3 &;
g[3,4]
51
g[u,v]
3 + u v^2
```

虽然纯粹函数概念的提出非常自然，但我们却有可能一直使用 Mathematica，而从来用不到纯粹函数．然而有时候 Mathematica 会把结果表示成纯粹函数的形式．因此有必要在这里简要介绍一下．有兴趣的读者可以在 Stephen Wolfram 的著作 The Mathematica Book 一书中或者 **Help** 菜单中找到关于纯粹函数的更多信息．

习题解答

A.1　把计算一个数的平方与它的平方根之和的过程表示成纯粹函数，并计算这个函数在 9 处的值．

解 ☞
```
f = #1^2 + Sqrt[#1]&;
f[9]
84
```

A.2　有一个数，是由两个数构造而来的，具体过程是把两个数的和的平方与它们的平方和加

起来得到的新数.把这个操作表示成纯粹函数,并计算当给定数为 3, 4 时的值.

解 ☞

```
g = (#1 + #2)^2 + #1^2 + #2^2 &;
g[3,4]
74
```

A.3 把函数 Sin 的导数表示成纯粹函数,并计算它在 π/6 处的值.

解 ☞

```
f = Sin'
Cos[#1] &
f[π/6]
```

$\frac{\sqrt{3}}{2}$

A.4 定义函数 $f(x) = (1 + x + x^2)^5$,并把它的 2 阶导数表示成纯粹函数.

解 ☞

```
f[x_] = (1 + x + x^2)^5
f''
```

$20(1 + 2\#1)^2(1 + \#1 + \#1^2)^3 + 10(1 + \#1 + \#1^2)^4$ &

A.5 把下述微分方程的解表示成纯粹函数,并计算它在 $x = \pi/4$ 处的值:

$$\frac{d^2y}{dx^2} + y = 0, y'(0) = y(0) = 1.$$

解 ☞

```
DSolve[{y''[x] + y[x] == 0, y[0] == 1, y'[0] == 1}, y, x]
{{y → (Cos[#1] + Sin[#1] &)}}
solution = %[[1,1,2]]
Cos[#1] + Sin[#1] &
solution[π/4]
```

$\sqrt{2}$

A.2 上下文环境

我们通常都是用名称定义符号的,从而达到可以重复使用这个符号的目的,然而有时候名称就显得有些不实用,使用起来很麻烦.所谓上下文环境就是在一次 Mathematica 操作中提供的帮助组织符号的工具.

实际上,符号的完整名称是由两部分组成的,即上下文名称与短名称,两者间用反引号(`)分开.因此反引号也称为上下文记号.

例 3 在下面的例子中,虽然符号 atomicnumber`au 与 atomicweight`au 有相同的短名称 au (金的化学符号),但它们是不同的符号.

```
atomicnumber`au = 79;
atomicweight`au = 196.967;
atomicnumber`au
```

```
79
atomicweight`au
196.967
```

当开始一次 Mathematica 操作时,缺省的上下文环境是 Global`,因此符号 obj 就等价于 Global`obj.可以通过重定义符号 $Context 的值来改变这个缺省设置.

- **$Context** 表示当前缺省上下文环境.
- **Context[符号]**返回符号的上下文环境.

例 4

```
au = "gold";
atomicnumber`au = 79;
atomicweight`au = 196.967;
au
gold
$Context = "atomicnumber`";          ←由于上下文名称是字符串,因此须使用引号.
au
79
$Context ="atomicweight`";
au
196.967
$Context = "Global`";
au
gold
```

内置 Mathematica 符号的上下文环境是 System`.

例 5

```
Context[Pi]
System`
```

经常见到在不同上下文环境中存在同样短名称的符号.如果只使用了短名称,那么 Mathematica 就会在称为 $ContextPath 的列表中确定它的位置. $ContextPath 的默认值为{Global`, System`}.

- **$ContextPath** 表示当前的上下文搜索路径.
- **Context[]** 给出当前 Mathematica 操作中所有上下文环境的名称列表.

例 6

```
au = "gold";
atomicnumber`au = 79;
atomicweight`au = 196.967;
$ContextPath
{Global`, System`}     ←这是默认的上下文路径.
$ContextPath = Join[ $ContextPath, {"atomicweight`"},
               {"atomicnumber`"}]
{Global`, System`, atomicweight`, atomicnumber`}
au
gold    ←Global`是 $ContextPath 中第一个元素.
```

```
Remove[Global`au]
au    ←现在 atomicweight`是 $ContextPath 中包含 au 的第一个元素.
196.967
Remove[atomicweight`au]
au    ←现在 atomicnumber`是 $ContextPath 中包含 au 的第一个元素.
79
```

在使用软件包中常见的一种错误就是在软件包上载之前就调用了命令或函数的名称.如果这时候再上载软件包,已经使用的命令或函数就好像不可再被调用.

例 7 函数 DaysBetween 包含在软件包 Miscellaneous`Calendar`中,它计算两个指定日期间的天数.

```
DaysBetween[{1999,1,1}, {2000,1,1}]                ←在上载软件包前就调用了函数.
DaysBetween[{1999,1,1}, {2000,1,1}]
<<Miscellaneous`Calendar`
DaysBetween::shdw : Symbol DaysBetween appears in multiple contexts {Miscella-
  neous`Calendar`, Global`}; definitions in context Miscellaneous`Calendar`
  may shadow or be shadowed by other definitions.
```

(符号 DaysBetween 出现在多个上下文{Miscellaneous`Calendar`, Global`}中;在上下文 Miscellaneous`Calendar`中的定义可能会覆盖其它定义,或者被其它定义覆盖.)

```
DaysBetween[{1999,1,1}, {2000,1,1}]
DaysBetween[{1999,1,1}, {2000,1,1}]
```

Mathematica 默认值仍是未定义的函数 Global`DaysBetween.

如果指定了完整的上下文路径,那么就会执行正确的操作.

```
Miscellaneous`Calendar`DaysBetween[{1999,1,1},{2000,1,1}]
365
```

如果去掉了 Global`DaysBetween,那么在 Miscellaneous`Calendar`中的函数就可以直接调用了.

```
Remove[DaysBetween]
DaysBetween[{1999,1,1}, {2000,1,1}]
365
```

A.3 模式

你一定已经注意到在 Mathematica 中定义函数时,要使用下划线(_).实际上,下划线的使用是 Mathematica 中的一个重要概念,称为模式匹配.

所谓模式就是类似于 x_这样的包含下划线字符的表达式.模式可以表示任何表达式.因此 f[x_]就是指定如何对任何参数值应用函数 f 的.当定义了函数后,如 f[x_] = x^2,那么就是告诉 Mathematica 只要有可能,就自动应用变换规则 f[x_] → x^2.

与之相对应的,对 f[x](没有下划线)的变换规则只是指定对文字表达式 f[x]的变换方式,而对 f[y],f[z]等则不做任何变换.

例 8

```
Clear[f]                    Clear[f]
f[x_] = x^2;                f[x] =x^2;
f[x]                        f[x]
```

```
x^2                                   x^2
f[y]                                  f[y]
y^2                                   f[y]
f[a + b]                              f[a + b]
(a + b)^2                             f[a + b]
```

x_ 可被任何表达式所匹配.	x 只能被 x 匹配.

例 9

```
1 + x^p + x^q/.x^q_->Log[q]
1 + Log[p] + Log[q]            ←所有的指数都变换成 Log.
1 + x^p + x^q/.x^q->Log[q]
1 + x^p + Log[q]               ←只有 x^q 被变换了.
```

通过模式可以指定表达式的类型以及格式. 例如, _Integer 就是整数模式. 类似地, _Rational, _Real 以及 _Complex 都是可接受的模式, 表示其它类型的数.

例 10

```
fact[n_Integer] = n!;              Factorial[5]
fact[n_Real] = "undefined";        120
fact[5]                            Factorial[5.5]
120                                287.555
fact[5.5]
undefined
```

例 11 在这个例子中定义如下函数:

$$f(x, y) = \begin{cases} xy, & \text{如果 } x \text{ 与 } y \text{ 都是整数}, \\ x + y, & \text{如果 } x \text{ 与 } y \text{ 都是实数}, \\ x - y, & \text{如果 } x \text{ 或 } y \text{ 中有一个是整数, 另一个是实数}. \end{cases}$$

```
f[a_Integer, b_Integer] = a b;
f[a_Real, b_Real] = a + b;
f[a_Integer, b_Real] = f[a_Real, b_Integer] = a - b;
f[2,3]
6
f[2.,3.]
5.
f[2.,3]
-1.
f[2,3.]
-1.
f[I,1]
f[i,1]              ←对于复参数值, f 没有定义.
```